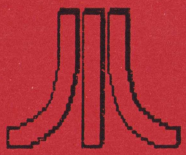
T—Y—N—E—&—W—E—A—R

TYNE & WEAR



ATARI (8)-BIT USER GROUP

Newsletter of TWAUG

Software
Editorial
Buy & Sell
Hardware

TYNE&WEAR
ATARI
USER GROUP

Reviews
Help line
Section
Repair Info

Public Domain Library

ISSUE #12

November/December 1994

U—S—E—R—G—R—O—U—P

BRING YOUR EIGHT UP TO DATE with power products from COMPUTER SOFTWARE SERVICES

THE BLACK BOX

The BLACK BOX is an add-on board for the Atari 600XL,800XL and 130XE 8-bit computers. It is a T-shaped board that plugs into the PBI port of the XL computer, or the ECI and cartridge ports of the 130XE. Connectors for both types of computers are built into the BLACK BOX so no adapter boards are necessary. A cartridge port is available on the board itself for 130XE users.

The BLACK BOX provides many unique and useful functions. The four primary functions are:-

* RS-232 serial modem port
* Parallel printer port
* SASI/SCSI hard disk port
* Operating System enhancements

The BLACK BOX is $199.95 for the basic unit, and $249.95 with an onboard 64K printer buffer.
Shipping and Handling extra.

THE BLACK BOX ENHANCER

A must for all BLACK BOX owners. The BLACK BOX ENHANCER is a plug in module for your BLACK BOX, enhancing the printer functions and adding an instantly available, full featured sector editor!

Installation of the BLACK BOX ENHANCER requires one simple solder connection. Only $49.95 plus shipping/handling.

THE FLOPPY BOARD

Our latest and greatest product. The FLOPPY BOARD is an add-on expansion board for the BLACK BOX interface. It allows the use of the same inexpensive floppy drive mechanisms used in IBM computers. The FLOPPY BOARD is the first floppy drive interface to support "high density" floppy drive mechanisms in either 5.25 inch or 3.5 inch. Built into the FLOPPY BOARD are our BLACK BOX ENHANCER and a version of our SUPER ARCHIVER to allow copying of protected disks for 3.5 inch format. Included with the FLOPPY BOARD is our program to read and write to IBM or ST formatted disks. This makes the FLOPPY BOARD the best way to transfer files to and from your 8-bit.

The FLOPPY BOARD is only $149.95 plus shipping & handling.

THE MULTIPLEXER

This device brings the power and flexibility of larger systems to your 8-bit. The Multiplexer is a collection of cartridge interface boards that allow up to 8 Ataris to read and write to the same drives (typically a hard disk), access the same printer(s), and talk to each other. It is the first practical networking system for the Atari 8-bit computer.

One "master" computer (any 8-bit) is equipped with the master Multiplexer interface. Then up to 8 "slave" computers hook up to this master, each having their own slave interface.

The "common" peripherals (things that are to be shared) are connected to the master. On each slave, all disk and printer I/O is routed through the master, so no extra disk drives are needed.

The Multiplexer sells for $199.95 for a master and two slave units with cable. Additional slave units are $89.95 each, plus shipping/handling.

THE SUPER ARCHIVER II

The SUPER ARCHIVER II edits and copies all enhanced density programs plus retains all the features of the SUPER ARCHIVER.

The SUPER ARCHIVER II is only $99.95 plus shipping & handling. NOTICE: if you already have THE SUPER ARCHIVER you may upgrade to S.A.II for only $29.95 plus shipping/handling. Software only.

THE BIT WRITER

The Super Archiver BIT WRITER is capable of duplicating even the "uncopyable" Electronic Arts and Synapse Syn-series, which employ 34 full sector tracks. The BIT WRITER must be used with the SUPER ARCHIVER

The BIT WRITER is only $79.95 plus shipping/handling.

THE ULTRA SPEED PLUS OS

The Operating System that should be in every XL/XE computer! The Ultra Speed Plus puts unbelievable speed and convenience at your fingertips.

Use any DOS to place Ultra Speed formats on your disks (with XF551 or modified 1050 drives), reading and writing at this speed with most programs. This high speed mode can be turned off for maximum compatibility.

Four simple solder connections are required for installation if your machine has a socketed OS ROM. The Ultra Speed OS is only $69.95 plus shipping/handling.

For more information on these and other 8-bit products:

CONTACT

COMPUTER SOFTWARE SERVICES
PO BOX 17660
ROCHESTER, NEW YORK 14617
USA

ORDERING LINE: (716) 429-5639
FAX: (716) 247-7158
BBS: (716) 247-7157

or contact T.W.A.U.G. we will do our best to help.

EDITORIAL

Who to blame!!!

John Matthewson
David Ewens
Max Gerum

We have had a request to use a countdown number system, similar to Page 6, to remind readers when to renew their subscription. David informed me that the database he is using gives him the renewel dates for each subscriber.

When a subscription is due David always encloses a new Order Form with the final issue, or a new form follows seperately shortly afterwards. David believes this is a better method to remind our readers than just a number on the mailing label.

The contribution fees for home and abroad:

HOME	1 COPY	£2.00
-DO-	6 COPIES	£11.00
EUROPE	1 COPY	£2.20
--DO--	6 COPIES	£12.50
ELSEWHERE	1 COPY	£2.50
---DO----	6 COPIES	£14.00

What a lovely response for PD software from our subscribers. We thank you very much indeed for your support and we hope to get some new software for your enjoyment in the near future.

REMINDER:

The book--The Complete and Essential Map--anyone unable to pay the full amount of the cash price can pay in instalments, whenever and whatever you can afford, at no extra charge. At the final payment the book will be dispatched.

The next issue will be ready by mid-January.

CONTENTS

EDITORIAL 3

MENACE II
Buying advice by Pimpf 4

ADVANCED MUSIC SYSTEM 5

PROGRAMMING THE ATARI
by David Davies 7

WORST & BEST ATARI SOFTWARE
by Richard Gore 8

ANIMATION TUTORIAL
by Ofer Saferman 10

WRITE PROTECT UNIT
by John Foskett 11

DISK CONTENT 13

CRACKING THE CODE
by Keith Mayhew 14

BOOK REVIEW
by Terry Chamberlain 17

CODE LOCK
by Len Golding 19

LETTER SECTION 23

GAMES REVIEW
by Mark Fenwick 23

GAMES COLUMN
by Mark Stinson 25

YORKY 256K UPGRADE
by Dennis Fogerty 27

APOLOGY & PATACHIN DOS
by TWAUG 28

AMS8 ADVERT
Bargains advert 29

ADVERTS FOR R.GORE 30
MICRO DISCOUNT 31
ACPC 32
CURRENT NOTES & PHOENIX 32

MENACE II

Or, a Buying Advice

By Pimpf - DBL (who else?)

First of all, excuse me for the poor title I put to this article, but you have to know that this is something like a continuation of what I wrote for issue #7 of MegaZine, that's why the title is a little clumsy. The first part was simply called "Menace!" and had some personal & common experiences within. The following text has also some of those experiences. Although I'm not the right one to give them, I hope my little advice will be somehow useful to you all, therefore this text will surely be published in (or almost sent to) some magazines & newsletters (MegaZine, The Best of Pokey Magazine, TWAUG Newsletter, etc)

Whether you're buying by mail or in person, there are some sensible precautions to bear in mind. Note that everything here applies to buying from businesses; most of it doesn't apply to buying from a private seller in any magazine's readers ad pages.

Always be absolutely clear about what's included in the price - postage & packing? Any necessary cables, etc? VAT? (where applicable - by law, and advert must say EXPLICITLY if VAT is not included; if it doesn't, VAT IS included. But this could be in small print, so read it all!) If you're buying in person, check that everything is there and it works properly before you leave the shop. If you're buying by mail, ring the supplier first (if possible) to confirm the price and availability, and ask what your options will be if there should be any problem. Find out when you can realistically expect delivery. Always keep all receipts and make records of all correspondence, whether it's by mail or by phone.

HOW TO PROTECT YOURSELF

Well, hasn't anybody told you how to do it yet? It's easy. Go to a pharmacy and ask for a pack of preservatives. Or if you wanna be exotical go to one of those Sex Shops and ask for a condoms catalogue, where you can choose from a rather big collection: from the 'Classic model' to those fluorescents ones to play "Star Wars" ::-)

Heeeeey! I think my multi-sided brain is going berserk a little bit... I'll beging again...

HOW TO PROTECT YOURSELF

Well, hasn't anybody told you how to do it yet? It's easy. If you are buying goods of more than -let's say- about 100 Pound Sterling in total value, always try to use a credit card. That way, in the unlikely event of anything going wrong, you'll be legally entitled to claim against the credit card company as well as the seller, even if the seller has gone bust. You may also get extra insurance - check with your credit card company.

If you're not paying by a credit card, pay by cheque. NEVER send cash through the mail - it's impossible to trace if it gets lost, you can't stop payment if you need to, and even if you have proof of posting you can't prove how much you sent.

Keep records. If you are paying by credit card keep a note of the exact time of the order and ask for an order number. If you are paying by cheque make sure that you fill in the details of date, amount and a payee's exact name in the stub - and keep it!.

WHEN A MAIL ORDER NEVER ARRIVES

At this point, read "Menace!" on MegaZine #7. Turn off your computer. Go to the Post Office and make some noise there with the help of your 3x3 bodyguards, some dynamite and a pair of Kalashnikov AK-47 to put things right! You may end up your days in jail but those suckers won't lose a pack anymore...:-)

Now seriously... Check everything carefully. If anything is missing, contact the supplier immediately. If something doesn't work, make obvious checks (the fuse, etc), but do NOT try to repair it!. If there's anything you're not happy with, don't use the product - if you do, you could be deemed to have accepted it.

IF THERE'S A PROBLEM

Whether you bought it by mail or in a shop, the law says a product must be:

1. 'of merchantable quality' (that is, broadly speaking, it must work)

2. 'as described' (in the advert or in person - and this one criterion also applies when you're buying from a private seller, not a business), and

3. fit for the purpose for which it was sold or for the purpose you specified when you ordered it.

If it fails to meet any or all of these criterias, then you are entitled to return the goods for a refund, receive compensation for all or part of the value, or get a replacement or free repair, depending on the seriousness of the defect. These are the 'statutory rights' that adverts always say are 'not affected' by any extra guarantees. The supplier cannot change or deny these rights.

If you are not happy about something, always contact the supplier first and politely explain your problem. Most problems are sorted out at this stage without fuss. If not, put your complaints in writing and send it to the manager or owner - ask for the correct name and job title.

Give the supplier a reasonable time to reply - say ten days or even the famous 28 days when your order something from overseas (i.e. the States).

If the problem is still not resolved, you may need to take some sort of legal action to enforce your rights. This need not cost you a lot (it also depends on the country you live in). Seek advice from a solicitor (ask about the free advice scheme) or from one of the following (check your phone book):

MENACE II continued

- Citizen's Advice Bureau
- Trading Standards Office (listed under your local council)
- Office of Fair Trading
- County Court (ask about the 'small claims procedure')

That's all for now. Till next issue!!

ADVANCED MUSIC SYSTEM

Have you ever wanted to show your fellow computer enthusiasts that your computer can do more than play games? Have you ever wondered if your computer's sound generators could do more than make explosion noises? Have you ever had the desire to compose your own music, but didn't want to learn an instrument? Have you ever been interested in electronic music but didn't want to spend several dollars on hardware? Or have you ever tried to use the Atari Music Composer cartridge, but given up in disgust due to its many limitations?

Well, no matter how you responded to the above questions, Lee Actor's ADVANCED MUSIC SYSTEM from the Atari Program Exchange is for you. The AMS is a music composing/editing/playing program that offers something for everyone. Both the program and the documentation are of superior quality, and the price is reasonable, making the AMS excellent value for anybody that owns an Atari with a disk drive. Purchasers of AMS will not find themselves singing the blues.

A BRIEF COURSE IN ELECTRONIC MUSIC

To fully appreciate the capabilities of the AMS, it is necessary to have an understanding of how electronic music works. When your ear hears a sound, it is actually detecting vibrations. The rate of vibration is called frequency, and determines the pitch of a sound.

In a musical instrument, a metal string, wood reed, stretched membrane, or air in a tube, is vibrating, and the player usually has a method for changing the frequency of vibration. Further analysis reveals that there are different kinds of vibrations. When viewed with an oscilloscope, vibrations have another characteristic called a waveform. Square, triangle and tooth are common waveforms.

Sound's other essential characteristic is volume. As a string is plucked or air is blown, The volume changes over a short period of time. This pattern of changing volume levels is called an envelope, and is usually divided into four parts called the attack decay, sustain and release. The attack decribes how quickly a sound reaches maximum volume, whereas decay tells how the volume fades away.

The frequency, waveform and envelope are all essential parts of a note, and a sequence of notes creates music. Elecronic music is merely a method of producing these qualities of sound by electronic means. A device which does this is called a synthesizer. Theoretically, it is possible for a synthesizer to imitate any musical instrument, or to produce sounds not available from conventional instruments.

A.M.S continued

The actual process is much more complicated than this brief description indicates. However, some individuals have made incredible accomplishments in the field of electronic music. The first significant commercial recording of electronic music was SWITCHED-ON BACH by Walter Carlos, now Wendy Carlos, who recently did the music for the Disney movie TRON and Stanley Kubrick's A CLOCKWORK ORANGE. Other notable names are Tomita (see SOFTSIDE, Nov.,1981.) and Synergy.

HOW ATARI COMPUTERS MAKE SOUNDS

One of the special chips inside your Atari is called POKEY, the POrt and KEYboard controller. The POKEY chip also contains four countdown timers which can act as sound channels. Each channel can have its own frequency, noise type and volume. A square wave is the only waveform available.

The noise type is sometimes incorrectly called the distortion, and there are six different types. Noise type ten is most commonly used and produces a clean, even tone. Other noise types can be used for white noise and other special effects. Sixteen volume levels, including "no volume" are possible.Each channel continually produces its sound, so duration must be controlled by the user.

For more information, see chapter seven of De Re Atari, or the Hardware Manual, chapter two, part B, and chapter three, part F.

ADVANCED MUSIC SYSTEM TECHNICAL OVERVIEW

The AMS is capable of playing up to four independent voices simultaneously, each with a range of five and a half octaves. The program uses the familiar noise type ten for the upper three octaves, and noise type twelve for the lower ones. The AMS can play up to 2400 notes per second, so it is capable of playing some very intricate parts. Each note can have one of fifteen different volume levels, or be at rest.

USING THE ADVANCED MUSIC SYSTEM

The version of AMS currently available from APX is an auto booting Atari BASIC program on a protected disk. The program uses a menu, reproduced here for reference, to direct all of its main activities.

A. PLAY
B. ENTER/EDIT MUSIC
C. ERASE
D. PLAY MEASURE(S)
E. SAVE
F. LOAD
G. DISK DIRECTORY
H. TEMPO
I. FORMAT DISK
J. AUXILIARY PROGRAM
K. AUTO-PLAY

The ERASE option should be used first to clear any previous music parts. The ENTER/EDIT MUSIC option uses the keyboard to enter new notes and edit old ones.

ADVANCED MUSIC SYSTEM continued

When this option is selected, one of the four parts must be specified, along with a measure number. Actual editing is done from the perspective of one measure. A note is entered by typing a letter key, A to G. Unless otherwise directed, the AMS will automatically insert appropriate sharps or flats according to the current KEY - one of the special editing aids.

As each note is entered, additional parameters can be included. These parameters include the octave, duration, whether the note is tie or abrupt, and the volume level. Valid octave numbers range from 1 to 6, the higher numbers indicating higher octaves. The range of the AMS is from C1 to F6, with C4 being "middle c".

The common durations for notes are whole, half, quarter, eighth, sixteenth, and thirtysecond. However, the AMS allows any imaginable note length to be specified. Sixty-fourth notes, triplets, septuplets, and dotted notes are all possible If no duration is given, the most recent one is used. A special error checking feature, the METER option, keeps track of the current time signature (4/4, 6/8 etc.) and warns you when there is a discrepancy between the time signature and the total duration of the notes in a measure.

Normally, when a part is played, there is a very short rest between notes to separate them. For a smooth, legato effect, the notes can be connected ("tied" or "slurred") by using the tie parameter. Conversely, the dot parameter shortens the playing time of notes, producing an abrupt, staccato effect. Rests are automatically inserted to compensate for the shortened playing times.

Aditional editing features allow the repetition of sequence notes or measures (a great time saver) and the ability to insert or delete notes and measures. While editing, it is possible to jump from one measure to any other, and any measure can be played immediately, making the AMS a very interactive music tool.

Overall, the author has done a fine job in creating a usable editor. Important information, such as the key, meter, total beats in the measure, and notes free, is clearly displayed at all times. (There are about 1400 notes available on a 32K machine, and 3400 on a 40K or 48K machine.) Errors are identified with messages. My only criticism of the editor is that it is too slow. This is understandable, since it is rather complex and is written in BASIC. However, I find that I sometimes want to enter notes faster than it can accept them. Persons who type using the one finger method are not likely to find this a serious restriction.

IT PLAYS THE PIANO

Without a doubt, the PLAY option is the best part of the programme. The graphics display is very good and reveals that the author has a firm command over the Atari's sounds and graphics power. The display consists of the black and white keys of a piano, with a total of five and a half octaves represented. As a note is played, the corresponding key is highlighted in one of four colours, showing the four voices. This is an extremely clear method of showing the relationship between notes typed into the computer, and the resulting sounds as they would be generated on a piano.

The PLAY option allows any combination of the four parts to be played, from just one voice to all four at the same time. Playing starts at the first measure. To play just a few measures somewhere within a whole piece of music, the PLAY MEASURE(S) options available. Normally tempo will have been previously set with the TEMPO option, but the paddles can be used to change the tempo while a piece is playing. Playing can be stopped at any time.

STILL MORE FEATURES

Disk oriented options include SAVE and LOAD, for storage and retrieval of music using DOS files, a FORMAT DISK command, and a handy DISK DIRECTORY function which also enables the user to lock, unlock, and delete files. Disk operations must always use drive number one.

The auxiliary program provides an alternate menu that lacks the editing option, but has recording and file merge functions. The advantage of this program is that it uses less memory, and with the MERGE function,allows the generation of longer pieces of music than is possible with the main menu program. The recording functions are for use with a suitable tape recorder. Through recording techniques, it is possible to extend the octave range and mix as many as sixteen voices together.

The AUTO PLAY program is a great way to show your friends what your computer can do. As the name implies, this program automatically searches the disk and plays every music file it finds. It will continue doing this until you stop it. Six demonstration music files are provided on this disk.

The AMS lacks a program to print music files to a printer, this could be a handy feature for some people. It is also important to note that the music generated by the AMS cannot be merged with other programs.

THE OTHER HALF OF THE ADVANCED MUSIC SYSTEM

Too often, documentation for a program is overlooked. Fortunately, however, the author's attention to detail is evident in the AMS manual as it is in the program. The manual is one of the very few to be granted a rating of EXCELLENT by the Atari Program Exchange. It is very complete - in thirty pages it fully describes all of the program's options, includes a sample session, and a section of helpful hints for entering music.

SOMETHING FOR EVERYONE

The AMS should appeal to a wide range of people. Because it is so interactive, the program is great both for the composer, or for someone just learning about music. The program is also a good demonstration of sound and graphics applications.

In order to enter music and understand most of the user manual, a minimal knowledge of music terms and theory is required. With the help of a knowledgeable friend, however,

AMS continued

a novice can learn the fundamentals and start entering music in just a few hours. For someone interested in learning music theory, may I suggest the book, INDEPENDENT STUDY IN MUSIC THEORY by Frederic Fay Swift. It is intended for younger students, but is complete, inexpensive, and can be read by anyone.

If you would simply like to have the computer play music, and do not have a need for the music entry/editing capabilities of the AMS, the program JUKEBIX #1 (APX 20135, 32K, disk, Atari BASIC, $15.95) might be of interest. Written by the same author, it uses the PLAY option of the AMS, including the piano keyboard display and tempo control by paddles, and comes with several demonstration pieces for you to enjoy.

THE ADVANCED - ADVANCED MUSIC SYSTEM

Although truly advanced, the AMS is still not the last word. The author is currently working on a revised version that will, no doubt, be the ultimate music program for Atari 400/800 computers. The revision will allow the program to run on a 16K, cassette based, Atari 400. The new version will be written entirely in Machine Code. The keyboard editing will be replaced by graphics oriented joystick editing, complete with musical staves. There will also be selectable envelopes. (The only way to do envelopes on the current version is to tie together several notes of increasing or decreasing volume.) Another major enhancement will be nestable subroutines, which, along with the other data compaction, is bound to reduce the size of music source files. Unlike the current version, the tempo will be changed by control of the music source, and different tone types should also be available. Other features are possible, but not yet definite. It will be many months, however, before the revision is completed, so persons interested in the AMS should get the current version now and start enjoying it,

THE GRAND FINALE

The ADVANCED MUSIC SYSTEM is a major step forward in Atari 400/800 software. The program is loaded with features, thoroughly debugged, idiot proofed, and properly documented. It is very interactive and fun to use. The graphics and sounds are most impressive. The many hours of development are evident, and it was time well spent. Finally, the program is priced right, making it a good software value. Needless to say, this reviewer is quite pleased with Lee Actor's ADVANCED MUSIC SYSTEM, and feels confident that users of the program will be pleased as well.

PROGRAMMING THE ATARI

By David Davies (ACPC).

Oh dear. I've a feeling I'm treading on dangerous ground here. After all, most Atarians use their computers to play games don't they? I mean, absolutely no-one follows these programming tutorial series, do they?

This is why this article is just a one-off. Just something to get you started ... if you really want to.

PROG THE ATARI cont.

I suppose some of you are already about to turn the page or put down this newsletter in order to have another go at 'Rescue on Fractalus'.

That's your decision, but you don't know what you're missing ...

Games players. Hmmm ... Too thick to program, eh? Don't know the difference between a PRINT command and your elbow? Programming a bit too much for you? Tsk!

No, I'm not goading you. Those are the sort of comments gamers are likely to come up with when someone suggests they try programming for a change. Comments like these are a load of %&#'!!

For a start, anyone can learn to program their Atari if they really want to do so. It's all a question of enthusiasm. I mean, look at me. I'm as thick as two short planks (hang on ... who typed that in? It certainly wasn't me!) and I can program in - deep breath - 6502 assembly, 68000 assembly, Atari Basic, STOS, Turbo Basic, Microsoft Basic II, GWBasic, GFA Basic, Turbo Pascal, PL65, Quick and a little COBOL. Phew! Hopefully I'll be picking up C++ soon as well! And to think it all started with a little sprouting of enthusiasm to get the most from my Atari some 8 years ago.

Okay, 8 years is a long time to learn all those languages, but once you've mastered Atari Basic then the sky's the limit. Okay, to guide you, here's my tip on the best path to take to the top of the programming tree ... assembly language:

Stage I:

High-level Interpreter

An interpreted language is one which converts each command line into machine code as the program is running. They are the slowest form of programming language but are generally easy to get into and ideal for beginners. BASIC is an interpreted language and you should first try to get to grips with this.

The Atari has several forms of the Basic programming language available for it. Here are my thoughts on three that I've used:

ATARI BASIC: Built into your machine. This is the obvious first choice for the beginner but is limited and you are bound to try Turbo Basic sooner or later. If you can, try Turbo Basic first since it has many more commands and, although it's to the most part compatible with Atari Basic programs, there are a few differences to some Atari Basic commands.

TURBO BASIC: An enhanced version of Atari Basic which is available in the public domain. It is advised that you ignore Atari Basic and jump straight into this if you can since it'll be less confusing in the long run.

MICROSOFT BASIC II: Available on ROM. This is a decent version of the standard Microsoft Basic which was commonly used in the early eighties on PC compatibles (I once saw an 8" disk version!).

PROGRAMMING THE ATARI continued

Many interpreted Basics have since employed many of its features. Microsoft Basic differs greatly from Atari Basic in many departments and uses what is now an accepted standard method in dealing with arrays and strings ... which Atari Basic does not.
In short, if you have another computer (like an ST) and would like to be able to program for that also, then you should consider trying to track down a copy of Microsoft Basic II - it's far easier to convert a program written in MS II to something like STOS than to convert one written in Atari Basic.

STAGE II:

Compiled languages

Compiled languages are those entered into some kind of a text editor and then the entire program is converted into a standalone machine code file. The drawback with compilers is that development time is much longer than with interpreters. However, the advantages are that compiled languages are often almost as easy to learn as interpreters and their resultant machine code programs are, on the best compilers, very near the speed of a program written using assembly language.

The Atari has more than its fair share of compiled languages. I've only used two of them so far but there are many more in the public domain, including some very good 'C' and 'PASCAL' compilers.

PL65: This, I'm afraid, is a bit of a rarity which is a pity when you realise just how good it is. This is my favourite language on the Atari and is a disk based compiler which was only available by mail order in 1987/88. I'd like to track down the original authors and have it re-published but I don't hold out much hope. Why is it so good? The language itself has very few commands but the authors thoughtfully provided a set of library files on the disk which are compatible with Atari Basic commands. This makes it an excellent next step for Atari Basic programmers. In addition, it can also give you a starting point in assembly since it has its own built-in assembler, thus allowing you to mix high and low-level code at will. A very flexible language indeed. Can produce some very fast code.

QUICK: This is the newest language that I know of for the Atari. Unfortunately, it isn't nearly as good as PL65 and is a three-pass compiler as opposed to PL65's single-pass compiler which, although works very well with short programs, can extend development time beyond even that of PL65 for long programs ... despite the fact that everything is memory resident. Not an easy language to learn despite the fact that the programmers have tried to produce some Atari Basic style commands. The problem is that it makes you think too much! For example, if you want to try and convert an Atari Basic program to QUICK and have a complex calculation to transfer, then you'll have to know something called the order of precedence in order to stop your variables getting out of hand. All I'll say is that if your calculations are very complex then you're in for a hell of a time! QUICK does produce some very fast code though and has some interesting features.

STAGE III:

Low-level languages

This is the very top of the tree. Once you've reached here then the world's your oyster and you can do what you like with your Atari. I won't bother with any descriptions here since there aren't any to give. There's only one low-level language to learn (unless you have a head for numbers and don't mind typing in a program in hex or binary!) and that's assembly language. Some assembly languages for the Atari include MAC/65, Atari Macro Assembler and the Assembler/Editor cartridge.

And there you have it. A very brief run down of some of the possibilities for you. Once you become proficient in Stage II then you could just stay there and never bother with assembly. But if you are proficient in Stage II then assembly language should pose little difficulty ... it actually took me two weeks to learn 68000 assembly language once I got on with it - 6502 assembly actually came a few weeks later!

There's no reason why you shouldn't at least give programming a try. You can get extremely frustrated if you're having difficulty with an odd problem (I've lost track of the number of times I've almost given up and blamed a problem on a bug in the language) but you get a real kick when you solve it.
Who the hell needs drugs?

THE WORLD'S BEST & WORST ATARI SOFTWARE.

Article by Richard Gore.

For those of you who don't know I regularly write articles for publication in DGS's Atari 8-bit News-Paper. Recently with the help of its readership I have written articles about the best and worst Atari 8-bit software. I asked for people to vote on what they thought were the best and worst pieces of software in various categories, however the response was pretty poor. For this reason I would like YOUR opinions, so I can get a better view of the general situation, any statistician will tell you the more samples are taken the more accurate the result should be.

There are seven categories I would like you to vote in, these are:-

WORST GAME, WORST PIECE OF SERIOUS SOFTWARE, BEST ARCADE GAME, BEST ADVENTURE GAME, BEST SERIOUS SOFTWARE, BEST GAME EVER and THE GAME YOU WOULD MOST LIKE TO SEE ON THE 8-BIT ATARI.

You should vote for your top three entries in all the categories, except for the last two categories which should have one vote each. All the nominations except for the last category should be software written for use on an Atari 8-bit computer.

The following are the results obtained from the News-Paper readership, please don't let these influence your votes.

THE WORLD'S BEST & WORST ATARI SOFTWARE. cont.

Vote for what YOU think, all votes will be handled in strictest confidence and you don't have to include your name if you don't want to.

WORST ATARI GAME

1. FRENESIS
2. CASTLE TOP
3. THE LAST V8

WORST SERIOUS SOFTWARE

1. ATARI MACRO ASSEMBLER
2. DOS 3
3. ALCHEMIST & SYSTEM 8 (joint third)

BEST ARCADE GAME

1. ZYBEX
2. INTERNATIONAL KARATE (joint)
3. STAR RAIDERS

BEST ADVENTURE GAME

1. ALTERNATE REALITY: THE DUNGEON
2. EMERALD ISLE & LEATHER GODDESSES
3. THE PAWN (my vote)

BEST SERIOUS SOFTWARE

1. TURBO BASIC
2. TRANSDISK IV
3. QUICK & SAM (joint)

BEST GAME EVER

ZYBEX & ALTERNATE REALITY: THE DUNGEON (joint)

MOST WANTED

SIM CITY

Well there you have it, (joint) means joint position holders for that spot. To be included on this list the software had to get more than one vote, and due to the small response only two games in the adventure section got more than one vote, so I made The Pawn third place because I like it. If you don't play adventure games please say so under the adventure games category.

Please send your votes as soon as possible either to the usual TWAUG address or directly to me at:

RICHARD GORE, 79 SPROTBROUGH ROAD, SPROTBROUGH, DONCASTER, DN5 8BW, ENGLAND.

I will do a follow up article in the next issue of TWAUG letting you know the results and hopefully doing mini reviews of some of the better or worst software mentioned. Thanks for your help, please respond.....

8-BITS FOREVER!

The President and Editor of the Garden City Atari Computer Enthusiasts in Victoria, British Columbia, Canada, has recently published a booklet entitled 8-Bits Forever!. It is a collection of humorous articles which have been published in various magazines such as New Atari User, Current Notes, STLOG, Pacific Yachting and West Coast Boat Journal. I am offering it for sale in Europe for £4, which includes airmail to England. I would ask for payment in postal money order and orders can be sent to:
Mr.Gordon F.Hooper, 253 Regina Ave. Victoria, B.C. Canada V8Z 1J6.

BASIC TUTORIAL - ANIMATION 2

This is the second part of the articles about animation. As promised in this part I'm going to discuss Character Sets and their use in animation.

The Atari, like any other computer, has at least one character set. A character set is a table in memory which tells the computer how to draw the letters, numbers and all the other signs the computer knows, on screen. The Atari XL has two character sets - the normal and the international, but this is just general information. The great thing about the character set is that the computer knows about it's existence by using a pointer to it, meaning there is a memory location that points to the character set. If we set it to point elsewhere, it can point to our character set. This is very useful because we can build our own set and then use it for different types of animation. Animating with characters is very similar to the page flipping concept we discussed in the previous article, but at a smaller scale, needing much less memory. The concept is to redefine some letters or other characters as we see fit, to a pattern, where every character is slightly different than the other, and then show them one after the other, thus producing animation.

The method definitely has drawbacks and we will get to know better ways, but for now lets get technical. First of all we usually don't want to redefine all the characters in a set (we will see later cases where we would want that) so we need some kind of copying routine to copy the original set from ROM to RAM. It can be done easily enough in Basic using PEEK and POKE but the character set is 1024 bytes long so it would be a tedious process. For that I have supplied a small machine language routine to incorporate in a Basic program, so it would do the job much faster. After we moved our set to a safe location in RAM (we will discuss later what is considered a safe location) we would like to redefine it. Each character in the set is built out of eight lines of data containing its form. Each line is built by adding the decimal values of all the binary locations of the lit pixels in the line. After this totaly obvious data maybe an illustration will help:

```
      The letter "A"
      ===============
|---------------------------------------|
|128| 64| 32| 16|  8|  4|  2|  1|
|---------------------------------------|
|   |   |   |   |   |   |   |   |   0
|   |   |   | x | x |   |   |   |  24
|   |   | x | x | x | x |   |   |  60
|   | x | x |   |   | x | x |   | 102
|   | x | x |   |   | x | x |   | 102
|   | x | x | x | x | x | x |   | 126
|   | x | x |   |   | x | x |   | 102
|   |   |   |   |   |   |   |   |   0
|---------------------------------------|
```

This way we build a letter or any other pattern we wish. (There is a small complication to this simplicity when attempting to build colorful patterns, but about that later).

The hard way is to sit and draw our patterns on a paper grid, which is very tedious, requires a little imagination, and is not very much fun so font editors were brought into this world to help us do the hard job, and I will include one with the article. With the use of a font editor we can redefine an entire character set and save it to disk as a 9 sectors font file which then can be loaded into our favourite basic program. Of course we can even turn our favourite font into data lines which can be poked into memory (but we have already agreed that this is tedious) or we can turn it into a string and copy it to memory using the above copy routine (details later). Now we have to discuss the real details of incorporating a font into a Basic program. The first thing we have to think about when we want to incorporate a font into a basic program, is a safe memory location for the font. The easiest way to allocate memory, that I can think of is to use 'MEMTOP(106)'. 'Memtop' points to the end of basic memory and it is used by Basic to allocate memory for the display list, the screen and other things.

If we lower the value of 'memtop' and immediately follow it with a graphics call then the computer will relocate memory for the display list and screen memory by using the new value, and all the memory from this value to the old value will be safe allocated memory for our puposes. Since one character set requires 1024 bytes of memory and 'memtop' holds the number of pages till end of Basic memory, we will have to lower its value by 4 (4*256=1024) and then do a graphics call (actually we have to always leave 4 pages of allocated memory unused because they're not safe).

Example:

```
CHBASE=PEEK(740)-4
POKE 106,CHBASE-4:GR. #GRMODE
```

This will allocate the memory as described. There is another way of allocating memory using MEMLO. If anyone wishes more details, write to me.

There is something more to be said. If you use a graphics mode with a text window do the allocation as above. If your mode doesn't have a text window, then the unsafe area after allocation is only one page so you have to POKE 106,CHBASE-1. If you plan to use graphics 8 and higher you have to allocate more memory thus: POKE 106,CHABSE-12. If you allocate memory for more than one character set and use graphics 8 or higher be sure that the total allocated memory would be a multiplication of 16. All this complication is necessary since when allocating memory this way some of it isn't safe and we must be sure that our needed memory is safe.

After we have allocated the necessary memory we will have to copy the font from a file or a string to our allocated memory. If you need to read the font from disk, I will include a very small routine which does the job. All is well and we have the font in memory, now lets use it. First we need to change location 756, which holds the old ROM value, to point to our new character set.

A small remark is in order. After each graphics call or opening the screen you have to change location 756 again because the computer restores the previous value.

BASIC TUTORIAL - ANIMATION2 cont.

After everything we did, and if you are still with me, we can use the font for animation. Take a look at the demonstration programs I included, and there you could see the principles I discussed. Program is in ARCed form on the issue disk.

It must be said that everything concerning character sets is very useful for other things and not only animation. For example, all the beautiful fonts in the quality games are done with redefined character sets, and many of the high resolution backrounds in many games use redifined character sets.

Lets go a little back. I mentioned a possibility to store the redefined character set in string and from there to copy it to memory. This is done by converting each data value to its ascii sign and thus storing it in a string. Then the string can be copied to memory using a copy routine and used as needed. This method is convenient when only a few characters are redefined, thus preventing the waste of 9 disk sectors to store the whole redefined set. It is also useful if you want to prevent an accidental erasure of a font file or if you want to prevent access to your redefined set from an external utility. There are programs to convert data lines or files to strings but I will not provide one this time. If anyone needs it or wants to write it by himself and has any problems, just write to me.

After we know more about character lets complicate life a bit and add color. There are 2 colorful text modes on the Atari XL - modes 12,13. Those are Antic modes 4,5 and are unavailable to Atari 800 users, but can be created by redefining the display list. In those modes it isn't enough to tell the computer if a pixel is on or off but also what color it is. This is done by using 2 bits instead of one. This means that for example in mode 12 the letters will be only half as wide as in mode 0. This also means that the characters have to be redefined a little differently. This is very hard using paper, but quite easy using the supplied font editor. All other aspects and principles we discussed remain the same. I will include a program to demonstrate multicoloured characters.

Now to conclude the article we have arrived at my favourite addition - the Turbo-Basic section. Life is much easier in TB. The copy routine which was discussed above is implemented in TB by the simple command:

```
MOVE from,to,no. of bytes (1024)
```

And the loading routine from disk is also fairly simple:

```
10 OPEN #1,4,0,"D:fontname.fnt"
20 BGET #1,memory location,no. of bytes(1024)
30 CLOSE #1
```

For character set animation, the 'move' command can be used but its a little tricky. The effort is worth it since the animation will be smoother and easier.

Some last remarks. First many of the supplied programs, and not only for this article, are not necessarily my own originally written programs.

If I know the author I will mention him or her, but usually I don't, and so as not to breach copyrighted programs I try to change each program even cosmetic changes. I claim no copyright and hope I haven't annoyed anyone by using his or her programs for demonstration and tutorial purposes. Feel free to change what you want, and to experiment with the programs, as it is the best way to learn. One last thing. If you feel I'm teaching simple nonsense, just remember to be patient till the fourth and concluding article of this series. And also remember that without the basic stuff you cannot get along.

If you have problems, questions, comments, criticism, on any subject concerning the Atari or computers in general write to me. My address is:

OFER SAFERMAN
21 BRANDE ST.
PETAH-TIQVA, 49600
ISRAEL.

See you next time in the third part of the Animation tutorial.

Bye for now.

ELECTRONIC WRITE PROTECT BYPASS UNIT

Designed and built by John Foskett

For use with a 1050 disk drive

September 1994

The standard "write protect switch" is fitted to the front of a disk drive to allow the user to write to side 2 of a disk, or to write to a write protected disk without having to notch it. Along with the switch are two LED indicators, one green and one red, the red usually flashes. The green shows normal mode of operation whilst the red indicates that the disk drives write protect system is being overriden or bypassed.

Since the switch actually replaces the drives own internal system, it clearly means that the write protect switch has to be switched to the "red" or system bypassed position to allow any form of a disk write to be actioned. It means therefore that every time any SAVE to disk is required, the switch must be in the "red" position. The "green" normal mode position can NEVER be used when writing to a disk.

The only time that the "green" position can be used is during a disk read and reading a disk is not a potentially destructive function. Because the disk drives "read" function is not linked to its write protect system, the position of the switch is irrelevant.

Because the write protect switch is irrelevant when a disk is read and can only be used in the one position (red) when writing to a disk, it then follows that there is no point in having the switch fitted at all. The write protect switch and its two indicators may just as well be thrown out and replaced by a wire link soldered directly across the

WRITE PROTECT continued

relevant connector pins, overridding the drives write protect system permanently once and for all.

Nobody would actually short circuit the system with a wire link, but call it a switch, especially "write protect" and it suddenly becomes the logical thing to do.

Nobody would dream of opening up their video recorder and disabling the anti- record mechanism to allow any cassette to be used for recording purposes, no matter whether or not the anti-record tab on the cassettes have been removed.

We would not do this since we would be afraid of recording over something that is precious to us, something we wish to keep. There would be nothing to prevent us from even recording over a rented film!!

The anti-record mechanism in our video recorders and the tab on the back of a video cassette is there for our benefit preventing accidental erasure. We are human beings and human beings do make mistakes.

I wonder just how many people have been lulled into a false sense of security believing that they have a safe method of writing to protected disks when in reality they have no write protection whatsoever.

I wonder how many people would have had the write protect switch fitted in the first place if they were known as WRITE DE-PROTECT switches which in reality is exactly what they are.

IS THERE AN ALTERNATIVE ?

Last year in June, I set out to design and build my own electronic unit since electronics is another of my hobbies. My electronic write protect bypass unit provides as much write protect safety as is feasibly possible whilst still allowing the system to be overridden. I have recently redesigned my unit to provide even more write protect safety, it now comprises of a much simpler and therefore a more reliable circuit.

The write protect bypass unit is housed in its own small plastic case measuring about 75mm long by 50mm wide and 25mm deep (approx. 3" x 2" x 1"). The unit sits beside the disk drive attached to it by a small lead, a bit like a mouse which means that no cutting or drilling is necessary to the disk drives case. The units lead enters the drive through a convenient ventilation slot in the lower section of the disk drives case.

The cable is connected to the drive via a miniature 4 way terminal strip which has screw connections eliminating the need for soldering. Only 4 connections to the drives circuitry are needed, all of which are to be found on the row of connectors along the lefthand side of the circuit board when looking into the drive with the cover removed. They are marked as J10 to J14.

The miniature 4 way terminal strip is allowed to float inside the drive at a convenient point between the aluminium heatsink and the outer casing with the units lead attached and passing through a ventilation slot.

From the terminal strip are 4 individual coloured leads to connect into the drives circuitry. The actual connections are achieved by inserting the leads into the top of the respective connector housing along side the connectors terminal pin. There is no need to unplug any of the connectors so no mistakes can possibly be made so long as the connections are exactly as stated on the diagram supplied with the unit. If required the units lead may be shortened by retracting it inside the drive and coiling it up rather than by cutting. If the units lead is cut, then care should be taken to ensure that the connections are correctly made.

The unit works in conjunction with the disk drives own internal write protect system which enables the normal use of the drive with a notched disk. It means that the unit may be left switched off whilst using a notched disk (side 1) and therefore reserving its use for an unnotched disk or a disk that is write protected. This ensures that the user does not get into the rythm of always using it.

The unit obtains its power requirements directly from the disk drives own power circuitry thus eliminating the need for a seperate power supply. Maximum power consumption for the unit is well within the capabilities of the drives internal circuitry. No power at all is comsumed when the unit is switched off.

On the front of the unit there are two LED indicators, one green and one red, an on/off switch and a touch plate for operating the unit.

Because the unit obtains its power from the disk drive, the disk drive itself must be switched on to enable the write protect bypass unit to be switched on.

Should the disk drive be open when the unit is first switched on, the green indicator illuminates to show that the unit is in the NORMAL mode of operation which means that the drives internal write protect system is NOT overriden. Whenever the unit is in the normal mode of operation (green), the user has full write protect safety at all times just as if the unit was not connected.

When the unit is operated by touching the "touch plate", the red indicator is illuminated which shows that the disk drives internal write protect system is being bypassed or overriden allowing an unnotched disk or side 2 of a disk to be written to.

The normal way of operating the unit is by its touch plate, but if a disk has been previously inserted into the drive and the drive is closed when the unit is switched on, the unit will then be switched straight into the BYPASS mode in readiness to write to the disk. This mode can be cancelled if necessary by simply switching the unit off and then on again to re-establish the safety of the normal mode of operation.

To ensure maximum write protect safety, the unit can only be operated when the disk drive is closed, thus any attempt to operate the unit whilst the drive is open is ignored.

WRITE PROTECT cont.

The unit will reset to the normal mode of operation when the disk drive is opened which ensures the full protection of any subsequent write protected disks should they be inserted into the drive. To enable a subsequent disk to be written to, the unit must be operated again.

For further safety, the unit will reset to the normal mode of operation after a time period of approximately 15 minutes to re-establish the full write protect safety should the unit be operated and forgotten.

Operating the unit by touch plate was incorporated since it is more positive than a pushbutton and it also reduces the risk of accidentally operating the unit by, for example, a misplaced book sliding or falling onto it.

The write protect bypass unit can only be operated by actually "touching" the touch plate with a finger (or any other bodily part).

To summarize, the write protect bypass unit has the following features....

1. Simple to install, no modifications required to the disk drive. It just plugs in.

2. Easy to use with the maximum write protect safety.

3. Powered from the drive eliminating the need for a seperate power unit.

4. Allows the normal use of the drives own internal write protect system.

5. Can only be operated with the disk drive closed.

6. Resets to normal mode upon opening the disk drive.

7. Resets to normal mode after a period of time.

8. Touch plate operation for positive operation and extra safety.

To obtain a write protect bypass unit apply to....

Mr. John Foskett
26 Auckland Road
Kingston-upon-Thames
Surrey
KT1 3BG (England)

The unit costs 12 pounds (inc P&P) and is supplied with instructions showing how to connect the unit to a 1050 disk drive.

Overseas, please add 1 pound P&P.

Please make cheques and postal orders payable to John Foskett and allow time for ordering components, assembling and testing the unit.

CONTENT

Side A of this issue 12 disk has another good selection of programmes for you to enjoy.

There is a game called CARGOSHP.BAS. You are in charge of five ship, the Cargoship is loaded with DRAM Chips from Taiwan. Your opponents have the same amount of ships to their disposal and they are trying to stop you from entering the harbour at Sanfrancisco Bay. Read the instructions of the game which are on the A side of the disk.

GIMINI: Test your solitaire strategy skills with this tricky tile game. Again I cannot tell you much about this game, some instructions are on the disk. Use the joystick to move the cursor to the tile you want to move and press the fire button.

SEARCHII.BAS: The documentation for this Database program is about four and a half pages long, I am telling you just in case you want to print out this text file that's on the disk. This database program is very flexible and easy to use. It is a MENU driven program with one key press commands.

When the program is loaded you are presented with a Main Menu, if no records have been entered choose (E) Enter Data. There is no need to set up the field size just type the data you wish to enter and to exit press return twice, you are now ready to enter the second record.

All your records can be sorted alphabetically, you can make hard copies of your data and search and well as view those files. Read the documentation to get familiar with some of the not so obvious features.

How are you in Geography? If you're not quite sure where all the countries are WORLQUIZ can help you. It is a very good education program for the young ones.

SPTIRE.EXE: This is a text adventure. In this adventure you're driving a car, whilst you're on the way to visit your mother's house. You swerve trying to avoid an animal, you loose control and crash into a hole. You are not hurt but you have a flat tyre. Read the text file for more information on this adventure, you'll find it on side A of this issue disk.

Side B of this disk is the ANIM2 tutorial program that goes with the text in the newsletter. You must unarc the program before use, you will also find the UNARC.COM program on the disk.

DRIVERS.ARC: Here are some 24 PIN colour printer drivers by Ralph Bradley, included in the arc file is all the documentation for the drivers.

There are a couple of DEMO programs all with doc files. One called JURASSIC.DEM and the other DLDEMO1.TBR and DTDEMO2.TBR.

MOVER.BAS: This memory mover program lets you move upto 256 bytes. An example program is on the disk also the doc file.

CRACKING THE CODE

by Keith Mayhew

Re-printed by M.Gerum

This article first appeared in "The UK ATARI Computer Owners Club" later renamed "MONITOR".

Part 12

CHARACTER MAPPED MODES

ANTIC mode numbers 2 through 7 provide the six different types of character display.

To recap, mode 2 is the standard text display mode which provides BASIC's GRAPHICS 0 mode. This offers a character cell eight scan lines high by eight pixels wide. The width of each pixel is the same as that in mode $F, GRAPHICS 8, (half a colour clock).

The colour scheme for this mode is exactly the same as that for the mode $F with a bit set to 1 indicating that the luminescence value of COLPF1 should be used with the colour of COLPF2 for that pixel, normally you would want to keep the luminescence value at maximum, 14 or 15 (bit 0 is ignored), so that character shapes appear virtually white.

Mode 3 is very similar to mode 2 but each cell is ten scan lines high.

Mode 4 has eight scan lines per cell and only four pixels in width. The data for each character is still eight bits wide but each pair of bits determine the colour of one of the four pixels, as in mode 3. Each pixel is one colour clock wide which makes each pixel twice the width of a pixel in mode 2 and 3.

Mode 5 is the same as mode 4 except that each pixel is made twice as high so that the whole cell takes 16 scan lines.

Mode 6, BASIC's GRAPHICS 1, has a cell 8 scan lines high by eight pixels wide with each pixel occupying one colour clock making this mode twice the width of mode 2. All the pixels in a character are the same, unlike modes 4 and 5 where four different colours can be displayed within each character.

Mode 7, BASIC's GRAPHICS 2, is the same as mode 6 but is 16 scan lines high.

CHARACTER SETS

Before discussing the six character mapped modes any further we will quickly look at the common factor between them; the character set.

Register CHBASE, shadow CHBAS, is used to tell ANTIC where the character set is stored. CHBAS forms the high byte of the address with the low byte taken to be zero. Each character shape is stored in the character set as a group of eight consecutive bytes for all modes with the first byte being used for the top most scan line of a character cell. In modes 2 through 5, there are 128 such characters in each complete character set, thus the set occupies 1K bytes of memory. Modes 6 and 7 have only 64 characters in their sets and so only occupy 512 bytes.

As with players, display lists and screens, you can have as many character sets in memory at once and swap between them by changing the value in CHBASE.

By using DLI's more than one character set can be displayed at once hence overcoming the limit of 64 or 128 characters for the mode being used. If you intend to make up many character sets then it is certainly worth having one of the character set generator programs which allow you to draw them on the screen with a joystick; it will be a lot easier and will save a lot of time.

BOUNDARIES

As with player/missile tables other data has to be aligned to certain boundaries in memory.

Character sets have to be aligned to 1K or 512 byte boundaries depending on which size character set is being used.

Display lists can start anywhere in memory but cannot cross a 1K byte boundary. Obviously it is best to start a display list on a 1K boundary so that you have the 1K maximum for your display list. Although it is possible to use the JMP instruction in a display list to cross a 1K byte boundary it has the side effect of creating a single blank line on the screen at that point. There is little use for this feature though, as all possible display lists will fit somewhere inside a 1K block.

The display data itself also has a restriction on where it can be placed. The Load Memory Scan, LMS, type instructions in a display list can cause data to be fetched starting from any location in memory, but as the data is fetched it must not cross a 4K byte boundary. It is often necessary to have more than 4K of screen data, e.g. BASIC's GRAPHICS 8 mode which requires nearly 8K. In such cases, the first LMS instruction is used as normal to start the display off, but when the 4K byte boundary is neared a second LMS instruction is used to re-load the counter to the next boundary. The problem this may now cause is that there is a gap in the screen memory at some stage, which can mean headaches when trying to calculate positions on the screen and their corresponding offsets into the display data. If this is a problem which you are faced with then a simple solution is to move the start of the display data so that the end of the 4K block corresponds to the end of one of the lines on the screen. The LMS instruction will then load with the next location, which is the start of the next boundary. By re-positioning the data in this way means that it can be kept contiguous, as if the boundary didn't exist.

If you are curious as to what happens if a boundary is violated the answer is simple; the data to be fetched after a boundary will actually be fetched from the start of the last block. That is they wrap around, analogous to when you increment a location from 255 to 0. As mentioned before, this was done to simplify the hardware design of ANTIC. It is something you will have to learn to live with!

DISPLAYING DATA IN CHARACTER MODES

The data stored in the screen memory for a character mapped mode consists of a series of bytes, each of which provide an offset into the character set to the appropriate character. A value of zero will display the character cell defined in the first eight bytes of the character set, one will display the next eight and so on.

CRACKING THE CODE continued

For example, BASIC's GRAPHICS 0 has forty bytes of data per line for each mode 2 instruction. Each value stored in this line represents the number of the character in the character set which should be displayed. The system has one character set which is stored at $E000 and extends to $E3FF, hence the value normally found in CHBAS is $E0. If you want to redefine fust a few of the characters from the system set then your program can copy the system character set into RAM and then alter the appropriate characters, and of course, finally altering the value in CHBAS to point to the RAM version.

As an example of this, Listing 1 shows some code which copies the system character set down into RAM starting at location $4000. It then copies in two new characters which have been defined on lines 580 and 590. Rather than copy these two characters into the first location in the character set onwards it skips over the first character, i.e. $4008 onwards. This is simply because the first character is usually a space, i.e. eight bytes of zeros, so that the screen can be 'cleared' by writing zeros to it; as the system does.

The program finishes by placing the code 1 into the top left location of the screen (pointed to by SAVMSC) thus displaying the first of the redefined characters and the code 2 in the position under it, i.e. at 40 bytes further on from the first. Lastly, the shadow character set pointer CHBAS is set to the high byte of the address of the new character set, i.e. $40.

The result of the program is to display a bad attempt at a capital sigma sign (a Greek character in case you didn't already know). Listing 2 is, as usual, a BASIC program to load the machine code. After running it, the code can be executed by typing:

X=USR(1536)

THE REST OF THE STORY

Firstly, it is worth remembering that the system character set is not stored in ASCII order, or ATASCII, as it is known on the ATARI. For instance ASCII space, which is 32 decimal, actually occurs as character zero in the character set. Why the character set wasn't made to match up with ATASCII, I haven't the faintest idea! anyway, the rule for converting ATASCII codes, which the operating system understands, into their character numbers is as follows:

ATASCII	Character
$00-$1F	$40-$5F
$20-$3F	$00-$1F
$40-$5F	$20-$3F
$60-$7F	$60-$7F

Thus when the O.S. is asked to print the upper case letter B, ATASCII $42. it has to convert it to $22 before saving the byte to the screen data so that a B actually appears. Note also, space, ATASCII $20, maps to $00 as mentioned above.

In ANTIC modes 2 and 3 the lower seven bits of a byte, select one of 128 characters. The eighth bit in these modes is usually used to give 'inverse video' characters, that is the fore and background colours are swapped over. The actual operation when bit 7 is set to 1 is determined by the register CHACTL, shadow CHART. Bit 3, if set to 1, causes all characters, regardless of mode, to be displayed upside down, whether bit 7 is set to 0 or 1. This can be useful if you want a screen which mirrors the top half the bottom half of the screen; you would copy the characters in reverse line order for the bottom half and use a DLI to flip bit 2 of CHACTL (not the shadow, of course) to a 1 reflect the characters themselves. Bit 1 causes characters with bit 7 set to be displayed in inverse-video. The cursor on the standard text screen is implemented by flipping bit 7 of the character under the cursor while bit 1 of CHACTL is set to 1. Bit 0 of CHACTL causes the character, if it has bit 7 set, to become empty, i.e. space. Thus if you have a string displayed in inverse-video and flip bit 0 of CHACTL on every few vertical blank interrupts then the characters will be flashing on and off. Setting bits 0 and 1 to 1 will cause a character to be displayed as an inverse-video space.

Mode 3 lines are displayed in exactly the same way as a mode 2 line except that an extra two empty scan lines are added at the bottom of each character. However, if characters are used from the last fourth of the character set, i.e. codes between $60 and $7F and between $E0 and $FF, then the two empty scan lines are displayed at the top and the eight character data lines below it. This mode was added primarily so that lower case letters could have proper descenders.

Modes 4 and 5, often referred to as multi-colour modes because they can have more than one colour inside a single character cell, allow upto 4 colours to be displayed in any one character. A pixel value of zero (from a pair of bits) causes COLBK to be used, 1 uses COLPF0, etc., as in the bit mapped modes which were discussed last time. Bit 7 is used in these modes to change the colour register used by a pixel value of 3. Normally, with bit 7 set to 0, COLPF2 would be used, but if bit 7 is set to 1 in a character then COLPF3 is used instead. Although there are still only four colours per character it is now possible to use a total of five colours for displaying character screens.

In modes 6 and 7 the lower six bits of a byte are used to select one of 64 characters. The top two bits determine the colour of the whole character. Values from 0 to 3 cause registers COLPF0 to COLPF3 to be used respectively. Where zero's occur in the character data itself, then COLBK is displayed.

SCROLLING

We will finish this time with an introduction to scrolling techniques; they will be covered in more detail next time.

to scroll data on most machines, including ATARI ST, the actual data in memory must be moved around, usually by the processor, to achieve the effect of scrolling text or background for a game. This is not only costly on the processor time but is rather a waste of time in itself. With ANTIC, scrolling is made easy and quick by changing the position of the screen relative to the data, rather than the other way a round. Thus to vertically scroll a screen, all that is necessary is to change the address after the first LMS instruction. For instance, adding forty to the address

CRACKING THE CODE continued

in BASIC's GRAPHICS 0 mode will cause data to be fetched from one line further on, thus the image will appear to have scrolled up the screen. It is up to you whether you prefer to think of the data scrolling up the screen or the screen scrolling down over the image!

Horizontal scrolling can be achieved in a similar way by just adding or subtracting one from the LMS address. Unfortunately, you will need an LMS on each line and to change each individually to do this properly, as we will see next time.

As you might have guessed such techniques will cause a coarse scroll of characters, i.e. a character at a time, or a pixel at a time for bit mapped modes. To scroll data more finely calls for the use of the registers VSCROL and HSCROL.

VSCROL, when enabled by setting bit 5 of a display list instruction to 1, causes that line to be scrolled up the screen by the number of scan lines specified. Thus with VSCROL set to 0 the line stays where it should, but as VSCROL is increased the line moves pixel by pixel up the screen. When the line is one pixel away from being at the next line's normal position, VSCROL should be set back to zero and a coarse scroll performed using the LMS.
HSCROL is used in a similar manner to shift a line pixel by pixel to the right, again using the LMS for coarse scrolling. The actual amount the data is shifted by is actually a colour clock, i.e. two GRAPHICS 8 pixels, as with players and missiles.

To fine scroll data in the opposite direction to that described above it is a simple matter of decrementing the value in the scroll register instead of incrementing them and changing the LMS's appropriately. fine scrolling can also be used in bit mapped display modes but their most common use is for scrolling characters.

Next time we shall look at how to put this theory into practice with a few sample programs.

Listing 1.

```
0100 ;Operating system shadows...
0110 CHBAS   =       $02F4   ;Character set base pointer.
0120 ;Operating system locations...
0130 SYSCHST =       $E000   ;System character set.
0140 SAVMSC  =       $58     ;Pointer to screen data.
0150 ;Page zero variables...
0160         *=      $CB
0170 SRC     *=      *+2     ;'source' pointer.
0180 DST     *=      *+2     ;'destination' pointer.
0190 ;Program equates...
0200 NEWCHST =       $4000   ;New character set address.
0210         *=      $0600
0220         PLA
0230         LDA     #SYSCHST&$FF ;System character set
0240         STA     SRC     ; pointer as source.
0250         LDA     #SYSCHST/256
0260         STA     SRC+1
0270         LDA     #NEWCHST&$FF ;New character set area
0280         STA     DST     ; pointer as destination.
0290         LDA     #NEWCHST/256
0300         STA     DST+1
0310         LDX     #4      ;Four pages to copy...
0320 NXTPAG  LDY     #0
0330 COPY    LDA     (SRC),Y
0340         STA     (DST),Y
0350         INY
0360         BNE     COPY    ;Copied page?
0370         INC     SRC+1   ;Next page.
0380         INC     DST+1   ;Next page.
0390         DEX
0400         BNE     NXTPAG  ;Finished all pages?
0410         LDX     #15     ;Copy sixteen bytes...
0420 CPYCHS  LDA     CHARS,X ;New character data.
0430         STA     NEWCHST+8,X ;Overwrite old data.
0440         DEX
0450         BPL     CPYCHS  ;Done?
0460         LDA     #1      ;Character 1.
0470         LDY     #0
0480         STA     (SAVMSC),Y ;Top left on display.
0490         LDA     #2      ;Character 2.
0500         LDY     #40
0510         STA     (SAVMSC),Y ;Under last character.
0520         LDA     #NEWCHST/256
0530         STA     CHBAS   ;Point to new char. set.
0540         RTS             ;All done.
0550 ;Data for two characters...
0560 ;When second character is placed under the first
0570 ; they (roughly) make a capital sigma sign.
0580 CHARS   .BYTE   $00,$00,$7E,$7E,$30,$18,$0C,$06
0590         .BYTE   $06,$0C,$18,$30,$7E,$7E,$00,$00
```

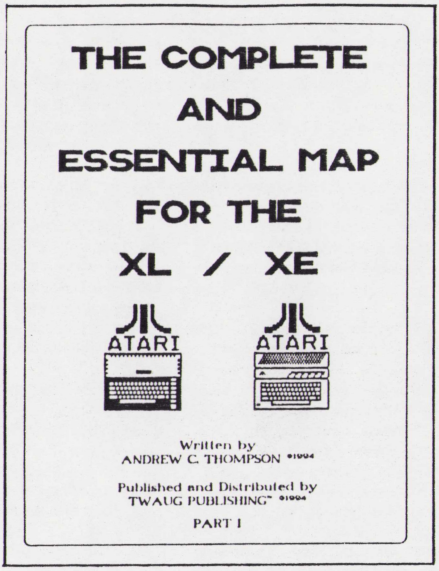
Listing 2.

```
QZ 10 DIM HEX$(16)
CV 20 LINE=10000:TRAP 100:J=0:START=1536
VA 30 READ HEX$,CHKSUM:SUM=0
AA 40 FOR I=1 TO 15 STEP 2
ZG 50 D1=ASC(HEX$(I,I))-48:D2=ASC(HEX$(I+
   1,I+1))-48
KT 60 NUM=((D1-7*(D1>16))*16+(D2-7*(D2>16
   )))
LN 70 SUM=SUM+NUM:POKE START+J,NUM:J=J+1:
   NEXT I
LY 80 IF SUM=CHKSUM THEN LINE=LINE+10:GOT
   O 30
IN 90 ? "Checksum error on this line:"
VO 95 LIST LINE:END
YS 100 PRINT "Data in memory."
YB 10000 DATA 68A900B5CBA9E0B5,1135
BS 10010 DATA CCA900B5CDA940B5,1077
SQ 10020 DATA CEA204A000B1CB91,1057
WY 10030 DATA CDCBD0F9E6CCE6CE,1732
FT 10040 DATA CAD0F0A20FBD4006,1086
JT 10050 DATA 9DB840CA10F7A901,864
IM 10060 DATA A0009158A902A028,764
AN 10070 DATA 9158A940BDF4A260,949
BD 10080 DATA 80007E7E3818C06,342
XB 10090 DATA 060C18307E7E0000,342
```

A SPOT OF "MAP" READING

Review of the BOOK by Terry Chamberlain

The Complete and Essential Map for the XL/XE, written by Andrew C. Thompson and published by TWAUG, is the first new book about the Atari XL/XE (or, in fact any of the 8-Bit machines) to be produced for many years. Even if this was its only claim to fame - and such is certainly not the case - Andrew Thompson is to be congratulated on his enthusiasm and perseverance in creating something of interest and use to his fellow Atari 8-Bit supporters.



OVERVIEW

So what has been created? The Complete & Essential Map comes as two A5 (6" x 8 1/4") ring-bound volumes, stretching to over 400 pages in total. Of these, the first 180-page volume comprises a full description of all the significant locations in the Atari XL/XE memory map.

This follows the same format as (and is based very largely upon) the Revised edition of 'Mapping the Atari' by Ian Chadwick - a book with which many of you will already be familiar. The second volume of the Complete & Essential Map contains a large set of appendices on a variety of Atari XL/XE subjects. These expand on many of the topics introduced in the mapping section of the book, and cover most aspects of operation and use of the Atari XL/XE computers.

The Complete & Essential Map is primarily intended to be a reference book for programmers, although it should also provide something of interest to anyone who is merely curious about how their Atari works and how it runs their various games and utility programs. While advanced programmers will undoubtedly find the book useful, it appears to be aimed especially at beginner and intermediate programmers, with the inclusion of many examples intended to expand their knowledge and abilities with BASIC, and to encourage them to try their hand at Assembler (machine-language) programming.

This is not really a book to be read cover-to-cover - although doing so is likely to teach you a great deal about what the Atari can (and cannot) do. It is a book to be referred to on subjects which interest you, and to be explored as you (hopefully) uncover the information you require. Every reader is almost bound to discover something they didn't know before. Andrew Thompson is obviously an expert in graphics and display programming. The sections of the Complete & Essential Map dealing with these topics, both as short notes in Volume 1 and in expanded form in the appendices of Volume 2, stand out as excellent.

As I said before, the Complete & Essential Map draws heavily on Ian Chadwick's 'Mapping the Atari'. Andrew Thompson has taken the three separate mapping sections of the original book, and brought all of the information relevant to the XL/XE machines together into a single volume. Whereas, in 'Mapping', to extract all of the information about a specific memory location you have to examine the main 400/800 listing, then check the addendum in Appendix 11, and finally check for any extra data in the XL/XE section in Appendix 12, the Complete & Essential Map has the relevant information gathered in a single place.

The author has also corrected a fair number of errors in the process, including many of the handler vectors and the memory usage of the standard display modes.

As well as being specific to the XL/XE series, it should also be said that this is a British book which, while not disregarding our transatlantic cousins and their NTSC systems, does emphasize how applications run with PAL displays - and manages to present a lot of new information in the process.

A particular feature of the book is the addition of short BASIC programs and routines to many of the mapping notes. These serve to illustrate the uses of memory vectors, hardware and software registers, and specific Atari XL/XE functions. The author claims that all of these little programs are fully tried and tested - unlike many of the sample programs found in 'Mapping the Atari'. Although I haven't tried them all out myself, everything that I did type in has worked as expected, so I have no reason to doubt his assertion. The best aspect of the example programs is that they provide a very good starting point for novice programmers, to experiment with the many features of their computer and develop their skills.

APPENDICES

The large number of appendices which make up Volume 2 of the Complete & Essential Map are a bit of a mixed bag.

BOOK REVIEW continued

They are collected into seven groups each of which deals, more or less, with a specific topic, although every group seems to contain at least one "wild" section which doesn't fit in anywhere else.

Appendix Group A covers the elements of Atari and Turbo BASIC, with a brief outline of keywords and their syntax, plus a good collection of hints and tips which should prove very useful to any level of programmer. Also included is a section on BASIC token values which, I feel, is unlikely to be of interest to anyone except an advanced programmer in rather specialised circumstances.

Appendix Group B provides a very good summary of sound generation with a set of example programs and routines covering the playing of music and sampled sounds. There is also a large section based on an article by Chuck Steinman on how to fit a second POKEY chip and hence allow your Atari to play in stereo.

Appendix Group C is the second largest appendix, (and probably the best) with 52 pages covering a variety of topics but without a single obvious theme. The first section presents the three sets of character codes used in the Atari together in one place for, I believe, the first time. This is followed by notes on the handling of numbers, addresses, and logic, and explanations of error codes. However, the best sections in this group provide explanations of graphics modes, player-missile graphics, and display lists, and are topped off by an excellent tutorial on display list interrupts (DLIs).

Appendix Group D deals with vertical blank operations, including vertical blank interrupts (VBIs), and the way the ANTIC chip controls the timing and structure of displays, again with particular reference to PAL systems. Display timing is a very tricky subject to deal with and, although his explanations get a little confusing at times, Andrew Thompson manages a good presentation from the point of view of a practical programmer. Included in this group is a section listing full details of all of the 6502 machine-code instructions, although without any examples of their use. This section also describes the effects of using the unallocated instruction codes - the 'illegal' op-codes. Here the explanations are, perhaps, a little too brief, and, although fair warnings are given, use of these codes in a program could (and the less-experienced programmer in some difficulties.

Appendix Group E is the largest in Volume 2 at 73 pages. It contains an abbreviated summary from 'Mapping the Atari' of XL/XE enhancements, and draws together most of the best known example programs dealing with copying ROM routines into RAM so that they can be customised. There is a corrected and expanded listing of DOS 2.5 memory locations, and a very brief (and not particularly useful) description of expanded memory management in the 130XE. However, the bulk of Group E is occupied with a source listing of the XL/XE Operating System, as disassembled personally by Andrew Thompson. The listing, with only a minimum of comments to identify the main sections, takes up all of 55 pages and is certainly a good start for all of us who really want to get to grips with the innards of our machines. Of course, what is really needed now is for someone to build on this and produce a fully-commented listing to make life really easy for the rest of us - a real opportunity for another book about the Atari XL/XE!

Appendix Group F gives brief details of the functions performed by each of the custom LSI chips (ANTIC, GTIA, POKEY, and PIA) which support and enhance the Atari's 6502 microprocessor, a few notes on 1050 disk formats, and pinouts of the various ports and connectors. The latter information is especially useful for being gathered together in one place, unlike in 'Mapping the Atari' where it is scattered throughout the text.

Last, but not least, Appendix Group G provides some comments on a few of the author's favourite programs, commercial and otherwise, a useful glossary of Atari-related terms, a list of suppliers still believed to be supporting the Atari 8-Bit, and an interesting collection of BASIC and Assembler sample programs. This collection, the final section of the book, covers a wide range of applications including graphics, music, special effects, and a database, as well as the disassembler used to produce the OS source listing in Appendix E.

COMMENTS

While I hope you gather from what I have written above that I have no regrets in buying the Complete & Essential Map (and I did buy my own copy - it was only later that David Ewens asked me to write this review!), it is not a perfect book and I do have a few criticisms to make.

Andrew Thompson's editing of the material taken from 'Mapping the Atari' is at times a bit too drastic. In quite a few locations he leaves out useful information which is still relevant to users of the XL/XE machines, and tends to cut explanations down to a minimum - sometimes to a level which could make it difficult for a beginner to follow. To be fair, some of the edited text has been transferred to the appendices, but in such cases a reference from the memory location to the relevant appendix would have been very handy.

A couple of examples will perhaps illustrate the point I am trying to make. Much of the explanation dealing with redefining character sets has disappeared from location 756 ($2F4) CHBAS, as have the concise descriptions and set-up details for graphics modes 9, 10, and 11 at location 623 ($26F) GPRIOR. The provision of sample BASIC programs at both of these locations is not really an adequate substitute for explanatory notes. Although bare details of the 130XE memory management still appear under location 54017 ($D301) PORTB, the explanatory notes in Appendix 16 of 'Mapping the Atari' have been replaced by a description in Appendix E3 which is unclear to the point of being almost incomprehensible to a beginner who has no prior knowledge of how the 130XE handles its memory banks.

I had also thought on first reading that the instructions for using the POKEY hardware timers in the notes at location 53760 ($D200) POKEY now lacked the key step of writing a value to STIMER at 53769 ($D209) in order to start the timer countdown operation. However, on closer inspection I found that Andrew Thompson has actually provided a complete

BOOK REVIEW continued

explanation of the process at location 532 ($0214) VTIMR4.

My other main criticism of the way the book is written is the tendency of the author to start on explanations of some topic and then give up part-way through. This happens, for example, in Appendix A3 where the topic of expanding or altering BASIC functions is introduced. Andrew Thompson then concludes that the process is rather difficult and leaves the reader, having whetted his or her appetite, high and dry. A similar half-hearted attempt at explaining the use of the joystick ports in Appendix F4 should also carry a health warning, since it encourages the reader to consider connecting miscellaneous electrical inputs to the ports - something you should only attempt if you know exactly what you are doing.

Despite the author's general success at correcting errors from 'Mapping the Atari', a few mistakes still remain. The bit values read from the joystick ports, location 54016 ($D300) PORTA, are normally 1 and go to 0 when the joystick is moved - not from 0 to 1 as stated. Also, in the program by Joe Miller listed at the next location 54017 ($D301) PORTB, the value in location NMIEN (54286 ($D40E)) is shown as being saved to the stack, and then restored at the end of the routine. However, as Andrew Thompson correctly notes elsewhere, NMIEN is write-only, and reading a value from it will always yield 255 ($FF) regardless of its true contents. Restoring this to NMIEN will enable both DLI and VBI interrupts which might not be at all desirable in all circumstances - programmers beware!

Andrew Thomson also perpetuates, in at least three places in the Complete & Esseential Map, the myth that the processor clock speed in PAL machines is much faster (at 2.217 MHz) than in NTSC machines. In fact, as reference to the Atari service manuals will reveal, PAL XL/XEs actually run fractionally slower than their NTSC versions, operating at 1.773 MHz (564 ns period) rather than 1.790 MHz (558ns period). With this single exception the comprehensive timing figures given in Appendix D2 are correct.

Finally, the index of memory locations at the end of Volume 1 seems to have ended up a little scrambled, with not all entries appearing in proper alphabetical order - a bit confusing until you realise this, and widen the search for the location you require.

CONCLUSION

Although I have made a fair number of criticisms of the Comlete & Essential Map - and it is always easier to criticise than actually create something in the first place - I do think that this is a book worth adding to your Atari collection. If you don't yet have any reference books then the Complete & Essential Map will make a good start.

No book is ever perfect for all readers or totally comprehensive, but this one certainly eases the task of locating data within the XL/XE memory map, and the original material and tutorial programs on graphics and display handling are alone worth the modest price. Nevertheless, I won't be throwing my copy of 'Mapping the Atari' away just yet!

A FINAL NOTE

For anyone curious about my credentials and qualifications for writing this review - I am an electronics engineer by profession, managing a large team of engineers and programmers who produce real-time computer systems for the Navy. I aquired my first Atari 800XL about nine years ago and my main interest lies in writing utilities, usually in Assembler. Although I must admit to owning a PC as well as a collection of Atari 8-Bit hardware, I much prefer the Atari for word processing (TextPro) and communications (BobTerm) - you feel so much more in control of the machine and its facilities.

CODE LOCK by Len Golding.

Originally written for the Atari User Magazine Sept/87.

Are you tired of people using your computer without permission? Here's a gadget that could be useful at home, school or at user groups.

It locks your computer by blocking the power supply until you enter a secret five-digit code. There's no key to be lost, stolen or copied. The choice of code number is entirely up to you and can be changed easily if the need arises.

The gadget fits neatly into your power supply lead. It doesn't require software and several fail-safe features are built in. It can control any Atari 8 bit computer, disk drive or program recorder - in fact anything that uses either a 5v DC or 6v to 9v AC power supply, at up to 2 amps.

Figure 1 shows the circuit. It's quite complicated, and you don't need to understand how it works in order to build and use it, but for electronic buffs here's the operating theory.

The lock must be able to detect when a digit is being entered, and discriminate between correct and incorrect entries. These three states are indicated by different voltages on the signal line: No digit = 2.5v, correct digit = 5v and an incorrect digit = 0v (or thereabouts). IC2a and IC2b are wired as comparators, to monitor the signal-line voltage.

If it is held at 2.5v nothing much happens. If it rises above 3.3v, IC2a sends a clock pulse to IC1. If it falls below 1.7v, IC2b sends a reset pulse to IC1.

IC1 is a decade counter with 10 outputs, though we use only five in this application. They are normally held low (0v), but go high (5v) in sequence from 0 to 9 when clock pulses are received at pin 14.

If the count reaches five, pin 12 goes from high to low, and we use this as success signal. Any incorrect digit along the way will generate a reset pulse (via IC2b) which sets the counter back to zero, wiping out the memory of anything previously entered.

The trick is to ensure that only the correct sequence of five digits can generate five clock pulses in succession. To see how this is done, let's trace the operation of figure 1, where the correct combination is 24579.

When you first switch on, C3 sends a brief positive pulse to pin 10 of IC2 and, via D1 and R1, to pin 12 of IC1. This does two things. First it latches the output of IC2c high, so TR1 turns on and RL1 pulls into its closed position.

CODE LOCK continued

No current can flow to the computer when the relay is in this state.

Second, it ensures that IC1 is set to zero, so pin 3 will be the only high output.

Now this output is connected to the S1 pin which corresponds to two on the dial. So if you rotate the dial to two, and press S2, the 5v at output 0 will appear on the signal line, generating the first clock pulse.

Output 0 now goes low, and output 1 goes high, so the second clock pulse will be generated only if you enter the digit which is wired to output 1 -- in this case four. This sends output 2 high, ready for the third digit (five), and so on, until the entire combination has been correctly entered.

When the count reaches five, IC1 pin 12 goes low. This drives the output of IC2c low, and R7 ensures it stays that way. TR1 turns off, so RL1 clicks into its open position. Current can now flow through the relay contacts, and the computer switches on.

Connecting the outputs of IC1 to different pins on S1 will program the lock to accept different combinations. Almost any five-digit code can be used -- the only restriction is that no digit may appear more than once.

There are a few components we haven't mentioned so far. C1 reduces electrical noise on the supply lines which might otherwise cause IC1 to count incorrectly. R2 and R6 produce reference voltages for IC2a and IC2b. C2 helps to de-bounce S2, and D2 protects TR1 from voltage spikes generated by the relay.
C4, BR1 and IC3 are needed only if you want to control 400/800 computers, disc drives, 410 program recorders or other items of equipment which use a 6v to 9v AC supply. The PSU for XL and XE models delivers 5v DC, so it can power the code lock directly.

Figure II shows the PCB pattern for those of you who like to etch your own, but you maybe able to get a ready etched and drilled PCB from a large electrical supplier. Holes for the terminal block and BR1 should be 1mm diameter, fixing holes are 3mm, and all other pads should be drilled 0.8mm.

Now let's look at the construction. Figure III shows the component layout for the version which operates with AC power supplies. If you're using an XL/XE power pack, omit BR1, C4 and IC3, and fit the two short wire links from A to B and D to E, instead of from A to C and D to F.

Several of the components are polarised, so they must be soldered the correct way round. BR1 has + marked, C4 has an indentation at the positive end, and the diodes have a coloured band to mark their cathodes.

Take particular care that IC1 and IC2 are inserted with pin 1 in the correct position, and that IC3 (if you're using it) goes with its flat metal side towards C4.

S1 and S2 are mounted on the case, and connected to the board via the terminal block. Figure IV shows how they are wired for the combination 24579. S1 is a 12-way switch, but the recommended control knob has only 11 positions marked (0 to 10). However, if you remove the fixing nut and shake-proof washer, you will see a small ring with a tooth which limits the spindle's rotation.

Prise this out and move it round until the tooth points at 11, then reinsert it. Now the switch will rotate to only 11 positions, matching the control knob markings.

The contacts on S1 are numbered, so it's easy to see where you are. One corresponds to zero on the control knob so, when you've chosen your combination, add 1 to each digit and solder a wire to the pin with that number. (That's why in figure IV, the combination 24579 appear to be wired for 3,5,6,8,10). The common pin is taken via S2 to the signal input, while all unused pins are connected together and taken to the reset line.

When everything is assembled, fit the unit into it's case, then attach the power supply input and output leads, the simplest, cheapest and safest way is to break into your existing lead.

You could fit an extra plug and socket but in this case remember that unless your power supply is well taped into the socket the gadget is pointless.

In AC mode, is doesn't matter which way round the two power leads are connected, so long as you get the input and output right.

Note that if you are using an XL or Xe power pack, the positive and negative leads for both input and output must be connected the right way round, or you will almost certainly damage the code lock, your computer or both.

Positive is the striped lead, negative is all black, and figure V shows the pin connections.

For security reasons, you might like to run a line of white insulating tape around the crack between the case and its lid. It won't keep determined people out, but at least you will know if anyone has been tampering.

PARTS LIST

		Maplin code
R1	33K orange/orange/orange	M33k
R2-R6	47K Yellow/violet/orange	M47K
R7	100K brown/black/yellow	M100K
R8	2.2K red/red/red	M2K2
C1-C3	0.1mid disk ceramic	YR75S
C4	330mid 25v exial	FB68Y *
IC1	4017BE decade counter	QX09K
IC2	LM324 quad op-amp	UF26D
IC3	uA7805 +5v regulator	QL31J *
BR1	W005 bridge rectifier	QL37S *
D1,D2	1N914 signal diode	QL71N
TR1	BC108C transistor	QB32K
S1	1 pole 12-way rotary switch	FF73Q
S2	push-to-make switch	FH59P
RL1	micro-miniature 6v relay	FM89W
	14-pin DIL socket	BL18U
	16-pin DIL socket	BL19V
	12-way PCB terminal block (optional)	RK74R
	plastic box (PBI (white)	LF01B
	control knob type NK2	RX01B
	pack 6BA x 1/8" spacers	FW33L
	pack 6BA + 1/2" c/s bolts	BF12N
	pack 6BA nuts	BF18U

* Not needed for XL/XE computers - see text.

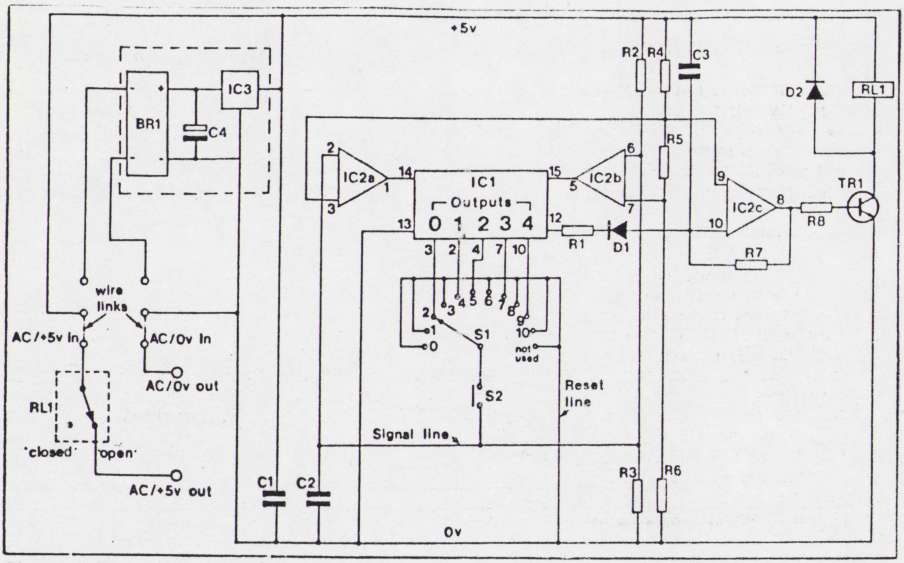


Figure I: The Code Lock circuit diagram

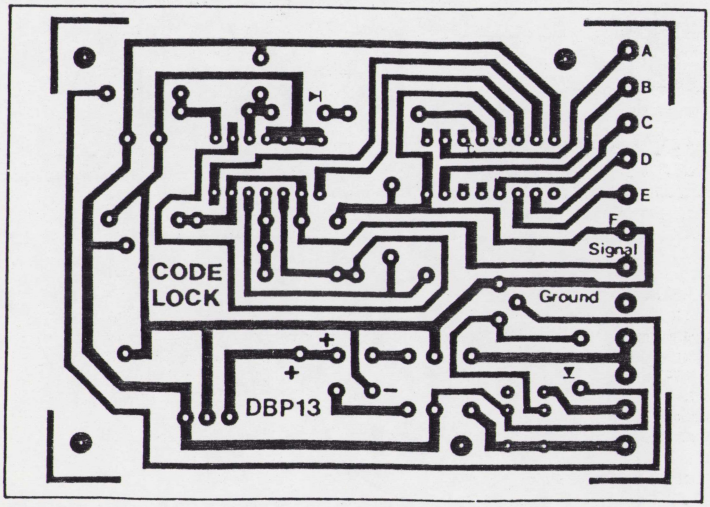


Figure II: The PCB pattern for Code Lock

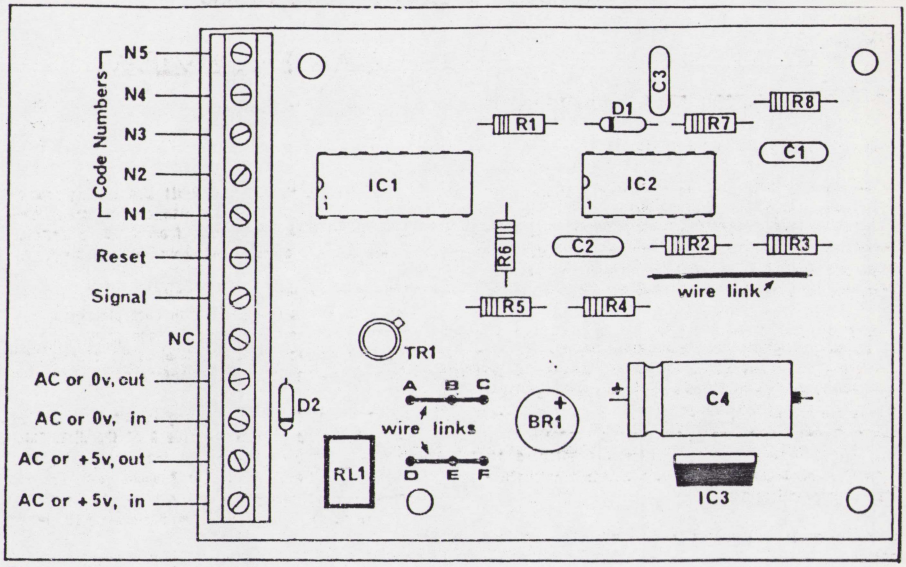


Figure III: Component layout

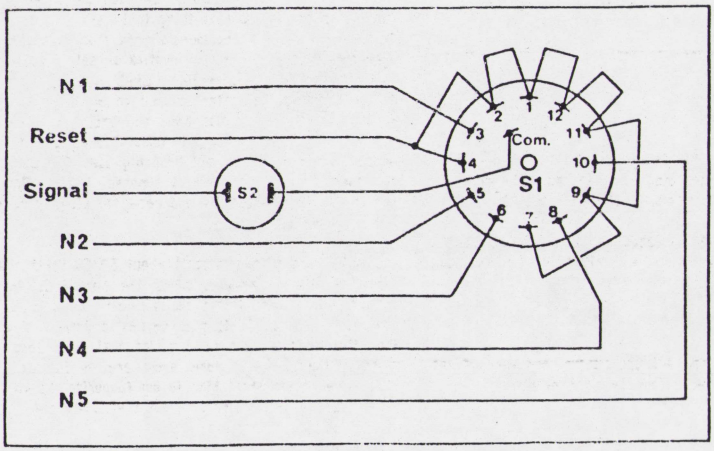


Figure IV: S1 shown wired for 24597

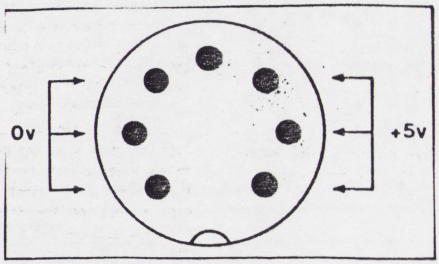


Figure V: View into plug fitted to XL/XE PSUs

LETTER SECTION

Dear Max/David/John,

In response to your reply to my letter printed in issue 11. May I say that I was not dissapointed in the size of the Complete and Essential Map. It is just that for some inexplicable reason I was expecting one large volume rather than two smaller ones. I fully accept your reasons for producing the work as it is, it was well worth doing.

Regarding Andrew Thompson's suggestion that the ROM OS could be replaced with a RAM OS for another machine in order to use its software. It has been pointed out to me that my comments about this could have raised unrealistic hopes about the feasibility of this idea. In short I have been told, by David Davies, that the difficulty of creating such an emulator, with the need to rewrite the software to take account of hardware differences, makes this approach a non-starter in practical terms.

I trust that this clears up any misunderstanding about the matter.

Yours sincerely,

Alan Hitchen

FOR SALE.

800XL, 1050 disk drive, 1010 tape deck plus software on tape and disk + mags. All boxed items and packaging in excellent condition £80.00 (inc. P&P) or may split. TEL. (0602) 720597.

800XL + 1050 disk drive all leads and power packs £60.00 (inc. P&P). Mapping the Atari (revised) £10.00 (inc. P&P). TEL. (0602) 720597.

FOR SALE

Mapping the Atari (revised edition) £10.00, and Micro Print Parallel Printer Interface as new £20.00. CONTACT:

Bill Jackson
68 Martin Street
Buckhaven, Leven
Fife KY8 1EP TEL.0592-716176

WANTED.

Any back issues of New Atari User, Page 6, old Atary User and Monitor. I have three 850 interfaces all boxed as new, I'm willing to come to some arrangement to exchange for mags. Please write to:

Mr. Nir Darey.
9342 Debra Ave.
North Hills, CA, 91343. U.S.A.

GAMES REVIEW

by Mark Fenwick

T34

Just when you thought your 8-bit was lacking new software, along come A.N.G. Software with another great game, T-34 The Battle. Like most games from A.N.G. a combination of strategy, great graphics and above all 'gameplay'.

T-34, a two player game, is basically a 'shoot your oponent before he shoots you' game, with each player controling a T-34 tank. However this objective can prove far from easy. With the mountain ranges and winds to contend with, each move requires a little thought before making your shot.

T-34 has one of the best loading intro's I've seen in a long while, taking up most of the side A of the disk. This can be bypassed by pressing 'start' as the screen prompt will tell you, but as it's so impressive I doubt you'll be skipping it for your first few plays. I'll not say too much about the intro so as not to spoil your fun, believe me it's great!

With the intro out of the way, the disk is flipped via a prompt and if you're a 130XE owner or have an upgrade you'll be told if any extra memory has been found and used. The options screen is the next thing to be seen, at first look it is somewhat bemusing as there's more than a couple! A mouse type cursor is used via the joystick to select various things including landscape, direction, wind, gravity and objects, though altering the latter seemed to do little to gameplay for some reason. You also have the choice to Quit or Play and clicking your cursor on 'About' takes you to a title screen, Why? don't ask me! Probably used for showing what the game title is at shows I suppose, hmmm. Once you're happy with your selections and ready to play, click on 'Play and away you go!

The main game screen is split about 50/50 horizontally, the top being the instruments while the bottom is the battle area. Between these is a prompts line (in English) for letting you know who's go it is plus sarcastic comments appearing from time to time to suggest you're taking too long to make a move. The comments vary, some are as follows 'Are you still alive', 'yeah shoot him' to my favourite 'Hasta la vista babe' now I wonder where they got that line from!

The instrument panel holds Power, for power behind shell fired, Angle to alter angle of gun turret, Fire to go to battle sequence, below is a Flag, which once pressed by either player conceeds that round. To the right are icons for scrolling the scenery and for moving your tank left or right. It's a good idea to select who's who player wise, as player one isn't always the one to start off. Before you alter the angle of your gun or power it's a good idea to see where your opponent is, as the only tank in view is the player about to go. This couldn't be simpler just click onto the scenery icon up/down, left/right to see where he is. Movement is very smooth as the foreground and background scroll at different rates and a sense of 3D is given. Now you have to judge the angle and power needed to carry your shell to successfully destroy your opponent.

GAMES REVIEW continued

Once happy with it, clicking on the bomb icon will remove the instrument panel to reveal the whole scenario and after a sampled firing speech, your shell is launched. The screen scrolls with your shell, though not always visible depending on the terrain. You may have to fire up off the screen and wait for it to drop to see just how lucky you were first time. A miss can result in a variety of sampled crashes and bangs. A miss will result in the instrument panel returning and your opponet having an attempt. A hit can result in various finishes, ie. The tank being totally wiped out, bouncing before exploding, or the tank exploding leaving the driver standing somewhat stunned by the whole situation! On each hit a score screen is shown from here you continue or return to options to alter things slightly. If you select wind, depending on the strength chosen it gives a whole new dimension to the game as your shell may need more or less power benind it, so as not to blow yourself up! A flag in the middle of the screen blows to show strength and direction, you'll certainly have a longer game this way, once you've mastered standard no wind first though.

I counted four scenarios in total, City, City ruins, Moon surface and Jungle though the latter could have been better in colour. Each one well detailed and colourful including a graffiti clad wall in the City scene. There are plenty of sampled sounds and some quite amusing exploding finishes for the tanks. Gameplay is very similar to that of the 'Snowball Fight' game especially where the winds are concerned, with your shell sometimes getting caught up in the wind and hitting your own tank. With the wind factor you need a bit of thought in prepering to make it count. The only sound part missing from T-34, which would have been clever and add realism, is a whistling sound as your shell falls back to earth, but all the same this has to be the best two player game since Megablast!

T-34 The Battle will be available from our Derek Fern priced hopefully in the £5.95 region and should be in good supply ready for this years AMS!

VICKY

Excellent intro's seems to be the order of the day with A.N.G.'s latest titles. This is soon apparent when booting up 'Avalons' Vicky the Viking, as Vicky's life and quest unfold.

On booting Vicky credits to author etc fade in and out like movie credits. A wizard like character is then seen sitting at a table with a thick mystical looking book, next to a slowly burning candle, below this picture are the words 'I will read you a story from an ancient book' As the wizard slowly turns the page the screen changes to show a camp scene, complete with burning fire in a moonlit sky, below which the old English text starts to flow. We're soon given an insight into who Vicky is and how on reaching his 20th birthday, is about to embark on his first expedition. After reading this press space continues the story as new text replaces the old. After each two pages of text the screen returns to the wizard turning the page from the old book. Well detailed pictures accompany each page of text to show the Viking ship through various scenes, from it's calm sailing across the sea to it's getting caught up in a violent storm.

This intro is very well portayed with several well detailed pictures adding an insight into the Viking's journey, with occasional returning shots of the wizard turning pages of the story book. The music is very well played and thankfully very nice to listen to, which is just as well as it plays throughout the whole intro.

In short the story tells of Vicky, a Viking no-less, on turning the age of twenty, accompanied by experienced warriors and advisors embarks upon an expedition. To prove his succession to his father as leader of the Vikings his expedition must prove a successful one. Soon after their voyage begins a violent storm errupts, lasting two days and the crew awake to find themselves in unchartered waters. Tired after the horrific storm they rest for a while, soon land is spotted, hungry for land they sail towards it. This land was supposedly deserted since the beginning of time, Vicky decided to stay there for a while. However, this soon turns out to be a grave mistake as a voice from the heavens sounding 'quite cross' called to the Viking's. It said of how angered it was of their arrival and that a task would be given to Vicky where 6 missions must be won to save them all. If unsuccessful they'd be banished to a hell between life and death. Gripping stuff eh!

The screen now changes to the play area to show your character 'Vicky' armed with an axe in the depths of the creepy caverns. The majority of the sreen takes up the main play area while a status bar at the base of the screen shows lives, cash and items picked up. You start off with ten lives, which soon depleat as you travel on your quest. The cash level refers to 'Denars' (funny enough, the very same currency of that sad sad holiday country of Tunisia, I should know I've just been there and never again, but that's another story!) which are awarded on killing certain nasties, this does appear to be random though, so my advise is to kill everything in sight. Items collected vary from a candle stick to a Magical Helmet which allows you to pass by the various enemies without being killed.

The play area is set out rather like a vertical maze, cut in to the stone with various exits and ropes to climb to higher parts. These graphics are well drawn with no blocky parts to be seen anywhere. The maze is massive inside as you may leave the screen at any height only to enter another part as the screen flips to another part of the maze. Because of it's size it would be a good idea to try and map your progress as some parts require re-entering to get to parts which seem impossible to reach at first glance. Vicky is well drawn and coloured in typical Viking dress complete with Viking helmet not forgetting his swinging axe. The nasties are various in shape, size and detail, from dragons to floating Jelly fish type creatures. These nasties are numerous and a short warning of flickering little stars warns of their appearance when a screen is entered, so be ready to wipe them out with a swing of your axe. Some will simply dissapear on contact with your axe, while others will leave a spinning coin 'Denar' in its place, these can be used later to buy various power ups described in the intro sequence.

Vicky the Viking is typical of most of 'Avalons' releases in its platform and pickups gameplay. It certainly offers a

GAMES REVIEW cont.

challenge judging by the size of the game and amount of enemies to destroy. The intro is excellent it really sets the scene, second only to the superb 'Aliants'. The music is great, with two different pieces, one for the into and a second for the main game, which can be switched of by pressing 'M' but I'm sure you won't do that. Though this part was reprogrammed into English by one of the A.N.G. team it would have been nice to have extended this translation technique to the main game, as all the screen prompts during the game are in Polish, which could at times leave you scratching your bonce!

Vicky will be available from Micro Discount along with T-34 and hopefully more new titles in time for this coming AMS. The price, hopefully, should be around the usual £5.95 region, which won't break the bank too much for quality 'New' titles like this!

MARK'S GAMES COLUMN

by Mark Stinson

AIR RESCUE

Air Rescue is an excellent budget game from Atlantis, featuring you as a helicopter pilot charged with the task of rescuing expatriate Britons from the midst of civil war. The cassette inlay informs you that 'peace has been shattered as the Military Junta moves against the popular uprising. Civil war rages, forcing expatriate Britons into hiding to escape the guns, tanks and missiles of the opposing armies. There is only one way to get them out - AIR RESCUE.'

The game is similar to one or two other games in which your task is to air lift survivors whilst avoiding enemy fire - most notably Choplifter by Broderbund. After having played Air Rescue for only a matter of minutes it became obvious that this one is by far the most well implemented version, and definitely worth buying even if you, like me, already own one of the other similar games.

Air Rescue has clear, colourful, well defined graphics, adequate sound effects, and a simplicity of play that makes the game a winner from the start. Small touches such as the ground shaking when the chopper is hit also add atmosphere to the game. There are ten levels of increasing difficulty, the latter of which will keep the most seasoned of games players challenged. Level 0 presents a screen which comprises a home base (the Hospital), trees, shrubs, houses and land based artillery. The object of the game is to navigate the helicopter near enough to houses to land and pick up the Britons whilst avoiding the scattered objects mentioned, including gunfire. Level 0 did not present much of a challenge, and after one or two collisions with trees and the like, I completed the rescue and moved on to level 1. At this point I wished I had stuck to adventuring! I found myself under fire from the artillery, from tanks, and from deadly missiles. On top of this, flocks of birds sent my chopper flying about madly in an effort to keep it in one piece. The game kept me coming back for more and I am now up to level four, in which the computer presents a strengthened force with much more fire power.

MARK'S COLUMN cont.

In summary, I would not hesitate to recommend this one to any games fan, especially at the price. You can't lose. Good news also for disk drive owners is that this one is Transdiskable - XTMENU, non standard read, default settings.

Title: Air Rescue
Publisher: Atlantis
Supplier: Page 6
Price: £2.95
Format: Cassette

THE TAIL OF BETA LYRAE

The Tail of Beta Lyrae is a game which is well known to most long standing Classic owners. It will run on all formats: 400/800 & XL/XE and is available from Page 6 on disk and cassette. The game was released back in 1983 by Databyte and is probably one of the few games most of us go back to time after time. The game is of a similar nature to the arcade game Scramble which was a huge success. Other variations are available, such as Air Strike. Of its genre, this is in my opinion the best of its kind. At 95p on cassette and £1.95 on disk it simply can't be missed.

The game story is well thought out, and instructions for play are given in the packaging and on the disk. The box is a slim plastic sleeve which helps greatly with neat storage.

The Tail of Beta Lyrae is set in a dust and plantetoid tail of the stellar twins Beta and Lyrae. The tail of the stars were a result of escaping gas and dust from the violent gravitational interaction of the twin bodies. The gas and dust combine with remnants of the system's inner planets, which, due to unstable orbits, had exploded eons before. The result was an outer tail teeming with planetoids of various sizes.

Colonists made settlements on these planetoids and mined them for precious metals which are to be found deposited in large quantities. In recent times a nomadic race of aliens have taken occupation of the tail, after continuous attacks on the colonists. Survivors are now terminated on site. As if you didn't know already, your job is to destroy the alien war machine and return the colonists to there rightful place.

In the game you play the part of a Galactic Wing Commander assigned to the Beta Quadrant. The mission will take you through a myriad of terrains, and will pit your wits against a plethora of different hazards. Your battle for each quadrant takes you through a mountainous terrain, through caverns, and over city settlements. If you survive for long enough, the computer will locate and take you to the alien power generators. With their power source at your mercy the aliens flee, and you get the opportunity to destroy their retreating fleet. You will then be taken to the next nearest alien presence. As you travel to the sector you will again need all of your skill to navigate through the debris of the tail.

The program allows you to choose from up to four players, five or seven wings (ships), and six levels.

MARK'S GAMES COLUMN continued

I have only ever played at the lowest level and I dread to think how difficult the higher levels must be. There are many targets to destroy during your mission, including communication antennae, alien vessels, plasma cannons, command centres, mining installations, lasers, missiles, asteroids, scanners and alien outposts. Points are obtained for destroying any alien installations and hardware, although the main task is to survive through the sections of the quadrant to find the alien power generators. Some great effects are employed in the game, with large explosions on destroying the bigger targets. You have to be careful not to destroy any target at too close a range though, or you may be destroyed by the debris.

The game is great to play, and has a high addiction factor, but it is by no means easy. Points are lost each time you fire, and it is very easy to blast away all your points very quickly early on in the game. Some of the obstacles presented during the game are very difficult to pass. For example, cannons fire vertically from the ground and are very often protected by mountains. To pass these particular nasties takes some considerable skill. Worse still are certain weapons which cast out a huge gravitational force which pulls you towards them at high speed. If you hit them, or any object in between, you are obliterated. There are also a number of indestructible barricades which can make navigation a nightmare. The gameplay is also very fast which makes each mission all the more challenging. The game makes good playing for novices and experts alike, and will just keep you coming back for more.

So, if you think you're up to the challenge strap yourself into the cockpit of your wing, put your pilot skills and warrior skills to the test and destroy the enemy. Remember, only a few have what it takes to make the mark, and you are the Galactic Wing Commander.

In the Tail of Beta Lyrae you get excellent gameplay, superb graphics, fast action, and good sound effects and music. Add to this the incredibly low price and it's a sure fire winner. Buy it now.

Title: The Tail of Beta Lyrae
Publisher: Databyte
Supplier: Page 6
Price: £1.95(d) 95p(c)
Format: Disk/Cassette

FOOTBALL
(THE AMERICAN VERSION)

Football is one of the later releases of Atari Corp. - RealSports TM. It is a simulation of American football, into which a great deal of material has been absorbed - mainly a very extensive set of rules and play formations.

Before playing Atari Football I knew very little about the game, other than that it appeared similar to rugby except that the ball is thrown forwards rather then backwards. Basically, the object is to beat the opposing team by scoring touchdowns in the End Zone, field goals and safeties.

The first two scoring methods are equivalent to a try and conversion in rugby. I am not completely clear on safety, but it has something to do with the defensive player furthest back on the field. Needless to say, there are a lot of rules to digest, especially if you know little or nothing about American football, but it is well worth the effort when you get to play the game.

Football is very unusual as games go. The difference with Football is that it is designed specifically as a two player game. There is a player v computer option, but it is not a full game, rather a practice forum in which you can learn set moves and the art of passing. The game therefore requires two joysticks, even in the player v computer option (you alternate between the two teams - the computer controls all defensive actions).

The game commences with options for a full game or practice game. Game length can be set at six minute or fifteen minute quarters. Once you commence play you must select your offensive formation out of fifteen variations. The defence then chooses its formation out of five variations. Then the action begins.

First the ball is 'snapped', or passed to the Quarterback. The Quarterback changes colour and you gain control of his movements. It is up to you to select where you wish to throw the ball, and your aim is to direct the ball to a runner at one or other side of the pitch. Once the ball reaches your other player (if it isn't intercepted) then your control switches from the Quarterback to the receiver. You must keep the ball moving towards the opposition's End Zone where you can score a touchdown or field goal. The defence will be relentless in trying to prevent this, and if you fail to advance ten yards in four plays, you lose the ball.

The instructions for the game are quite comprehensive, and even include a playcard giving details of offensive/defensive plays and tactics which is placed between the two players. The game is certainly novel. However, in order to get the most out of it you must have a partner to play with.

Title: Football
Publisher: Atari
Supplier: Page 6
Price: £7.95
Format: ROM

NEW ATARI USER PAGE 6

The only magazine left in this country that supports the 8-bit There is a large Public Domain Library available. Your support is needed to keep the magazine going.

For more details write to:

PAGE 6
P.O.BOX 54
STAFFORD ST16 1TB

THE YORKY 256k UPGRADE.

By Dennis Fogerty.

Something had to be done. Despite, eventually, acquiring a second 1050 Disk Drive, output was becoming very time consuming. With demands for print work ever increasing from friends and family, most of my time on a 800XL was spent using just two pieces of software, namely, Textpro 4.54+ Word Processor and Daisy Dot III (commercial program).

DD III is a wonderful print processor with an ability to mix numerous text fonts, but when each one has to be accessed from disk the drive works overtime. This is especially the case when using the marvellous magnified fonts. Add to this a personal quirk that, while I am a good proof-reader with a hardcopy in my hands, I fail to spot spelling and formatting errors on-screen. So, when my drives were not busy with printing, they were loading Textpro for corrections then back again to DD III for re-printing. At times I marvelled the drives didn't blow up!

I did consider a 130XE with its RAMdisk facility. But that would have meant finding room for 2 computers because, no way would I throw out my trusty 800XL.

Then, an advert appeared in Page 6 for the YORKY 256k Plug-in Memory Upgrade, but the price of £50 plus £2 postage caused some hard thinking. However, after saving like mad, I had an informative telephone chat with the advertiser, Richard Gore and, finally, made my decision to buy.

Waiting for that parcel to arrive, the old saying 'Fools and their money are soon parted' often came to mind. I need not have worried though. I have had my YORKY for a year and to part with it now would bring tears to my eyes!

YORKY occupies very little space behind my computer. It's simply a grey, plastic box some 6ins long x 3ins wide x 2ins high. From one side extends a 49-way ribbon cable fitted with an edge connector which simply snaps into the parallel Bus socket. At one end of the case is a 3-way toggle switch, a 7-pin DIN socket, and a short length of orange cable terminating with a 7-pin DIN plug.

The computer's power supply cord is simply removed and inserted in YORKY's DIN socket. YORKY's orange cable then takes its place in the computer, and the 3-way toggle switch now takes over from the Atari's on/off switch. In the central position, YORKY and the computer are switched off. Moving it upward selects the normal 64k memory which, I understand, was retained because a few software programs fail to run when 256k is selected. Toggle downward and the 800XL's memory has expanded to 256k.

My only criticism of this hardware, (and it hardly seems worth mentioning), is the orange cable. It has a diameter of a quarter of an inch, (similar to the extension cables used on garden mowers!), and is quite bulky and stiff. Had it been a thinner, more flexible, cable my YORKY would stay in the position I choose. As it is, it seems to have a mind of its own.

The Support software supplied is on a DS/DD disk but a Single Density version is available for those with 810 drives. The demo files ably demonstrate what can be done with increased memory - most impressive. But what pleased me most was the modified Atari DOS 2.5 on that disk. Booting this DOS in 256k mode provides a 707 sector RAMdisk in D8: with Dup Sys and Mem Sav installed. In addition, a full 707 sectors are available in D7: after formatting manually.

What a godsend! Now I have Textpro and several text files in one RAMdisk and DD III, together with an endless number of fonts, in the other. After copying everything into RAMdisk, I can recall either Textpro or DD III in about three seconds flat. Compared to pre-YORKY days, when loading either program took nearly 1 minute, this was a revelation. Also, printing now continues without a break. When the text file calls for a change of font, there is no longer a delay waiting for a disk drive to load it into the computer.

Although I guess Atari 2.5 DOS is the most popular Operating System, SpartaDOS 3.2 and MyDOS both provide massive RAMdisks, while SuperDOS 5.0 automatically provides a 1011 double density sector RAMdisk on boot up. I quite like SuperDOS 5.0 but find it corrupts Textpro so I'm back with Atari 2.5. I understand, though, that SuperDOS works well with AtariWriter Plus.

I have found YORKY to be compatible with a lot of 130XE software. MyCopyR copies single sided disks in one pass, just like the 130XE. Transdisk IV, AtariWriter Plus, Megablast and Top 3 Demo all work perfectly. Other programs I use are the Graphic Art program, Draw7, and Diskcat 130XE.

Disk Copier/Sector Editors abound in the Public Domain but there is one, Ram-Edit 130XE, which reads an entire Single Density Disk into RAMdisk. Sectors are then read instantly to the Editor instead of having to be read from disk, saving not only time but wear on the Disk Drive. Incidentally, Ram-Edit 130XE is one of many files on the Page 6 PD Library Disk#154 "Hackin' About".

That is my personal "hands-on" experience of YORKY after 12 months intensive use. My vedict? - INDESPENSABLE!. Yorky is available from Richard Gore, 79, Sprotbrough Road, Doncaster, DN5 8BW. Telephone (0302) 784642.

TWAUG
8-BIT

T.W.A.U.G.
P.O.BOX No.8
WALLSEND
TYNE & WEAR
NE28 6DQ

APOLOGY

In issue 11 we had an article on digitising photos. Unfortunately the BASIC program that goes with it was missed off the disk. To rectify that mistake we've included the program on this issue disk.

Please accept our apology for this mistake.

PATACHING DOS 2.5

On the previous page is an article on the upgraded XL to 256K, by Dennis Fogerty. He also says that he's very pleased with the modified DOS 2.5 that gives him two RAMdisks of 707 sectors.

This program first appeared in the MAAUG newsletter and is by Kurt Grittner.

INSTRUCTIONS on how to modify DOS 2.5

1) Insert your assembler cartridge.
   Put disk with DOS 2.5 in drive 1 (without RAMDISK.COM) Cold start your computer.

2) When the "EDIT" prompt appears, go into debug mode by typing: "BUG".

3) When "DEBUG" prompt appears, go into mini-assembler by typing: "A".

4) Type the following lines ending each line with the RETURN key. The mini-assembler will respond to each line with one or more lines of assembled object code.
NOTICE - You must type a space after each "<" sign.

```
B86< LDX #0
105C< LDA #$64
12DE< LDY $21
< CPY #8
< BCC *+4
< ADC #5
< TAY
< LDA $185B,y
< JMP $1846
148D< CMP #6
14C4< LDA #$70
< JSR $1846
183A< .BYTE "MEM.SAV ERR",$9B
1846< STA $44
< LDA $D301
< AND #$83
< ORA $44
< STA $D301
< RTS
185B< .BYTE $20,$24,$28,$2C
< .BYTE $40,$44,$48,$4C
< .BYTE $60,$64,$68,$6C,
77F< CPX #7          (change # for other
< BCC $78A           (drive number
B88< CMP #7          (-2,3,4,5, or 6.
< BCS $BD3
105E< CPY #7         (Don't type this!)
< BCS $1076
70A< .BYTE $C3
```

Continued next page...

PATACHING DOS 2.5 continued

5) Press RETURN to EXIT the mini-assembler.

6) Type "X" followed by RETURN to return to the EDIT mode.

7) Type DOS to access the DOS Menu.

8) Write DOS files to disk.

9) Turn off the machine and cold start DOS...with RAMDISK.COM on the disk. When you do a directory of D8: you should have DUP.SYS, MEM.SAV and 620 sectors free. Use the "I" Option to format D7:. When you do a directory of D7: you should have 707 sectors free.

All Micro Show

Going to the show?

Please come and see us at our stand and say HELLO.

We have plenty of software bargains.

Plus we have 5 800XL's upgraded to 256K for sale, all with a power unit.

PC XFORMER 2.0

What is a PC Xformer...XF2 is software that emulates the Atari 800/800XL on a PC. This software is available from our stand, it is a set of three disks two 5 1/4 and a 3 1/2. This is PD Software.

PC XFORMER 3.0 the commercial software which will be released shortly it emulates the upgraded 800XL and the 130XE. It allows for full bank switching and much, much more.

SIO2PC

This is a combination of hardware and software, this software is on the set of disks mentioned above. The hardware is an interface that consist of a box with two cables running from it. One end you connect to your Atari I/O port and the other end to a PC serial port. Basically, this turns the PC into a peripheral for the Atari. You can use anything hooked up to the PC for the Atari as you would any normal Atari device. You can save files to the PC floppys or hard drives, you can send to the printer hooked to the PC and you can even use the PC as a RAMdisk for the Atari. This gadget works with any IBM compatible computer. The cable is easy to make, but should you have any difficulty in making the device get in touch with T.W.A.U.G., we are pleased to advice and help.

SEE YOU AT THE SHOW.

ATARI SUPPORT from RICHARD GORE

ARENA: The full 50 level version of the excellent puzzle game that was a demo bonus on Page 6's issue disk.
Price: £5 ($10) XL/XE Disk only.

BUBBLE ZONE: A fast blast em set on a grid being invaded by ever expanding bubbles. Collect the tokens and see how far you can advance.
Price: £5 ($10) XL/XE Disk only.

JAWBREAKER: Classic maze game based on Pac-Man, munch the sweets but don't forget that dental hygene! Two different mazes supplied which were originally separate games.
Price £4.50 ($10)

MOUSEKATTACK: Another classic maze game, but this one involves you plumbing all the levels of Rat Alley. Addictive arcade action with a simultaneous two player mode.
Price: £4.50 ($10)

Jawbreaker and Mousekattack are available as a double pack for only £6.50 ($12). They are available on disk, tape and Rambit turbo tape, please state which when ordering.

YORKY 256K plug in memory upgrade with manual and support disk.
Very few units left. Hurry this could be your last chance!
Price: £50 ($90) +p&p (UK £2, EC £3.50, world £5)

OTHER COMMERCIAL SOFTWARE.

I also have limited stocks of older commercial software on a periodic basis. All titles are new and still shrink wrapped with original (English) documentations. At the moment I have the following titles available:

Disks at £4.75 ($10) each: SILENT SERVICE, HALLEY PATROL, TAXICAB HILL, GORF, WINTER CHALLENGE & SUMMER GAMES.

ROMS at £6.95 ($14) each: GATO

All software prices include p&p to UK, overseas please add £2 per order unless otherwise quoted. Payment by cash, cheque (payable to Richard Gore) or IMO. Prices in dollars are for USA customers and include shipping but payment must be made in US dollars cash, preferably using registered post!

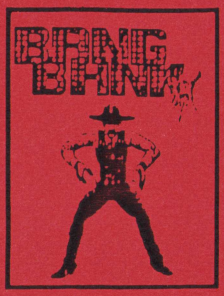
Contact address:

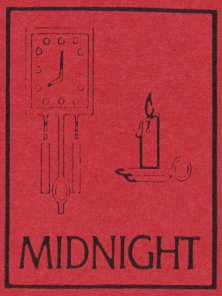
RICHARD GORE, 79 SPROTBROUGH ROAD, SPROTBROUGH, DONCASTER, DN5 8BW, ENGLAND

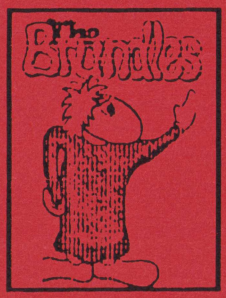
Telephone: (0302) 784642

Plus coming soon..... new software from Germany, GTRACKER (a new sample sequencer), Golf Tour 94 (a new golf game), Super Print-LAB XE and possibly (in the UK) the ICD/OSS product line such as SpartaDOS X, R-Time 8, Action!, MAC 65 etc.

NB I will be at AMS 8, helping out on the DGS stand, watch out for some great offers. I hope to see you there...





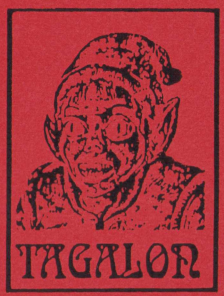


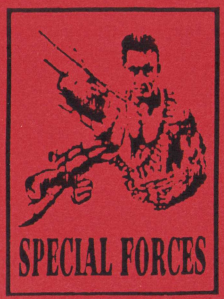
Are You Still Complaining about lack of New software for your Atari 8 bit ?
If you are you don't know about Mail Order From

MICRO-DISCOUNT

265, CHESTER ROAD.
STREETLY.
WEST MIDLANDS.
B74 3EA.
ENGLAND.

TEL 021 353 5730
FAX 021 352 1669





ADAX *HANS KLOSS*

DARKNESS HOUR

The Atari Classic Programmer's Club

Would you like to become a life member with access to various programmer services, a helpline, regular printed newsletter and discounts off our software? If so, then we suggest you hurry. The life membership option is due to be abandoned after Christmas and all subsequent new memberships will be on a 12 months subscription basis only.

Current UK membership cost is:

12 months: £6.00
Life: £12.00

Overseas members most welcome. If you require more details then please send an SAE to the address below. Overseas Atarians, please send two International Reply Coupons (available from your post office) for more information.

Still Available:

Swift Spreadsheet (Standard) - £9.95

Swift Spreadsheet (New) - £12.95

(NOTE: New version includes a revised 40 page A4 manual. For details of the standard version and of the program itself, please refer to the review in issue 67 of New Atari User).

Available Soon

Menu Print: Still under development and currently undergoing its THIRD rewrite! We apologise for the delay but we expect the programming to be definitely completed by the end of September. More details will be published when ready.

THE ATARI CLASSIC PROGRAMMER'S CLUB
Pen-Tyddyn
Capel Coch
LLangefni
Anglesey
Gwynedd LL77 7UR
Wales

CURRENT NOTES

Helping Atari Owners Through the World of Computing.

Current Notes is published monthly (excluding January and August), in the U.S.

This magazine or you can call it newsletter has 80 pages full of computer news and very good articles that covers the Atari 8-bit and ST. It comes in full size of 11 inches by 8 1/2 inches.

SUBSCRIPTIONS:

Europe subscriptions is $69 per year.
Bankers drafts made payable to Current Notes
to: CN Subscriptions
122 N. Johnson RD.
Sterling, VA 20164

NOTE: VISA and MasterCard accepted.
Call (703) 450-4761

Editor's Note:
We at TWAUG are receiving the CN magazine monthly and we are always looking forward to it.

PHOENIX.

The new disk based news letter from Ireland, produced by Robert Paden.

PHOENIX a double sided disk, side 'A' will be packed full of text files containing Articles, reviews and much much more. Side 'B' will contain a good selection of PD software.

PHOENIX is only available from Robert Paden himself.